DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/ENG/91D-23

GABOR SEGMENTATION OF HIGH RESOLUTION SYNTHETIC APERTURE RADAR IMAGERY

THESIS

Michael Alan Hazlett
Captain, USAF

AFIT/GE/ENG/91D-23

Approved for public release; distribution unlimited

AFIT/GE/ENG/91D-23

GABOR SEGMENTATION OF HIGH RESOLUTION SYNTHETIC APERTURE RADAR IMAGERY

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Electrical Engineering

Michael Alan Hazlett, B.S.E.
Captain, USAF

December, 1991

Acknowledgments

I wish to thank all of those people who made this research possible. First, I would like to thank my advisor, Dennis Ruck, for all his advice and patience in those times when things weren't going as planned. Second, I would like to thank Steve Rogers for his classes on pattern recognition. These classes peaked my interest in the area of pattern recognition and led to this research. Finally, I would like to thank my wife, Carol, and my children, Katie, Alex and Emily for all their love and support during the stressful time of preparing this research.

Michael Alan Hazlett

Table of Contents

List of Figures

List of Tables

Abstract

This thesis investigates the use of Gabor filters and a radial basis function (RBF) network for segmentation of high resolution (1 foot by 1 foot) synthetic aperture radar (SAR) imagery. Processing involved correlation between the SAR imagery and Gabor functions. Two methods for selecting the optimal Gabor filters are presented. This research used complex Gabor functions and operated on single polarization HH complex data. Following the selection of the proper Gabor filters, correlation coefficients for each image were calculated and used as features for the RBF network. Provided are results demonstrating how Gabor processing and a RBF network provide image segmentation.

GABOR SEGMENTATION OF HIGH RESOLUTION SYNTHETIC APERTURE RADAR IMAGERY

I. Introduction

1.1 General

Research in pattern recognition at the Air Force Institute of Technology (AFIT) has been working toward building an autonomous target recognizer (ATR). The eventual goal is to incorporate this ATR in a missile capable to autonomously locate and destroy a hostile target located within the missile's field of view.

To date, many techniques have been developed which provide a limited solution to the problem of object or texture recognition by machines. Complete machine recognition remains an unsolved problem. There currently exist machines capable of limited recognition within a controlled environment. These exist in supermarkets and industrial production lines. The environment for these machines often control noise, object scale, orientation, and position so the features are easier to extract.

The development of a target recognizer is much more complex. These machines will typically be housed in small spaces such as missiles, aircraft or spacecraft. In addition, the environment will no longer be controllable thereby creating a whole new set of processing problems. There will be conditions of extreme noise and temperature along with limitations imposed by the processing speed of the computer and the algorithms used.

The ability of a target recognizer to be able to pull out cultural items, such as targets, requires obtaining good sensor data. Assuming that high quality sensor data is available, then the three steps in the target recognition process are segmentation, feature extraction and classification (12). Segmentation places a label on each pixel in an image and locates blobs of interest. Feature extraction computes a number of features for each blob detected by segmentation. Classification assigns a label to an input feature vector generated. This research will focus in on the first step of the target recognition process.

1.2 Background

Previous work by Captain Albert L'Homme (6) used Gabor functions and an Artificial Neural Network (ANN) to segment Synthetic Aperture Radar (SAR) imagery. His approach correlated the image with a set of nine Gabor filters in the frequency domain. He then used the coefficients of the Gabor filters as input vectors to train a Radial Basis Function (RBF) to segment SAR images. The success rate he achieved in segmenting was better than 80%. However, his work had a few limitations. First, only a limited set Gabor filters was used. Second, only the magnitudes of the SAR data was used, therefore, all phase information was lost. Finally, only a single polarization, horizontal-horizontal (HH), of the four polarizations of data available was used. Any information contained in the other polarizations was lost.

1.3 SAR Imagery

The data used for this research was collected using a high resolution SAR sensor. The data has been provided by the Massachusetts Institute of Technology Lincoln Labs and was collected during testing of the Advanced Detection Target Sensor (ADTS). The data has a resolution of 1 foot in azimuth and 1 foot range and is stored in complex format (in-phase and quadrature). The imagery is of a New England farm area and consists of trees, fields, shadow, and cultural features (farmhouse, barn, etc,). A later set of data collected from the same area will also be used. This later data has been processed by a polarimetric white filter in order to reduce the amount of speckle in the imagery. A description of the filtering process can be found in (8).

It should be clarified that the shadow regions which appear in the imagery were created by regions of no return and not as a result of the ambient sunlight. A sample of the SAR imagery is shown in Figure 1.

1.4 Problem Statement

There does not currently exist a robust computer method for segmenting high resolution SAR imagery.

1.5 Research Objectives

This thesis will:

Figure 1. Sample ADTS SAR Imagery: Mission 85, Frames 27-30

- Search for an optimal set of Gabor functions associated with regions of trees, fields, shadows, and culture through the use of histogramming and Fourier techniques.
- Use the correlation coefficients from the optimal Gabor filters as inputs to an ANN which will segment the SAR imagery.

1.6 Definitions

Segmentation is the process of dividing up a scene based on the image structure to identify objects of interest from background regions.

Pattern recognition is the process of determining the correct identity of the objects of interest segmented from an image.

Radial basis functions are a type of feedforward neural network that compute by using neurons with local receptive fields. These neurons encode the inputs by computing the closeness of the input to the center of their respective receptive fields. (16:II- 133)

Correlation is a measure of similarity between two objects.

1.7 Scope

The major focus of this research will investigate which combinations of Gabor filters will provide the best ANN input feature set in order to segment SAR imagery into regions of trees, field, shadow, and culture.

1.8 General Approach

Initially, the work to be performed will include some 256 different variations of the Gabor function used to process the complex SAR imagery. Next will be selection of the optimal Gabor filters. Two methods for selecting the optimal filters will be examined. The first method selects the optimal filters based upon the filters with largest coefficient for a particular region type (trees, grass and shadow). The second method, suggested by Bovik (2:63), will perform a Fast Fourier Transform (FFT) on selected regions of the image and select the Gabor filters whose frequency response matches with the two strongest peaks in the image power spectrum. The coefficients from the selected filters will in turn be used as vectors to input into an ANN for the purpose of training the ANN to segment other SAR imagery.

1.9 Overview

Chapter II presents a review of current literature applicable to Gabor filtering and radial basis functions as related to segmentation of high resolution SAR imagery.

Chapter III presents methodology used for this research. Provided is a description of Gabor functions, techniques for determining the optimal Gabor functions, and a description of the ANN used for image segmentation.

Chapter IV discusses the results and analysis of this research.

Chapter V presents conclusions and recommendations for further research.

II. Literature Review

2.1 Introduction

This chapter reviews literature found useful for application of Gabor filtering and radial basis functions. The intent is to find techniques, within this literature, which can be used for segmenting high resolution SAR imagery. In addition, the literature will be used to show the validity of using Gabor filters and radial basis functions combined as a method for segmenting SAR imagery.

2.2 Gabor Filters

2.2.1 Biological Motivation Work by Jones and Palmer (5), Daugman (4), and Porat and Zeevi (10) has shown that the two-dimensional Gabor functions accurately fit the receptive field profile for simple cells in the striate cortex of mammalian vision systems. As an illustration of the closeness to which the Gabor functions match the meat, in Figure 2, the top row of graphs are the measured receptive fields of simple cells in the striate cortex of cats. The second row of graphs gives the best fit Gabor functions. Finally, the bottom row of graphs show the difference between the measured data and the calculated Gabor functions. Notice there is very little difference between the two sets of graphs, and what difference there is, appears to be random noise.

Gabor functions are simply a Gaussian envelope modulated by a complex sinusoid of the form given below.

$$G(x,y) = \exp^{-\pi[(x-x_o)^2\alpha^2+(y-y_o)^2\beta^2]} \exp^{-2\pi j[(u_o(x-x_o)+v_o(y-y_o)]} \quad (1)$$

Where: x_o, y_o are position shifts.

u_o, v_o are modulation parameters and determine the orientation of the function as: $\Theta_o = \arctan(\frac{v_o}{u_o})$.

α, β are scale parameters for the Gaussian envelope in the x and y directions, respectively.

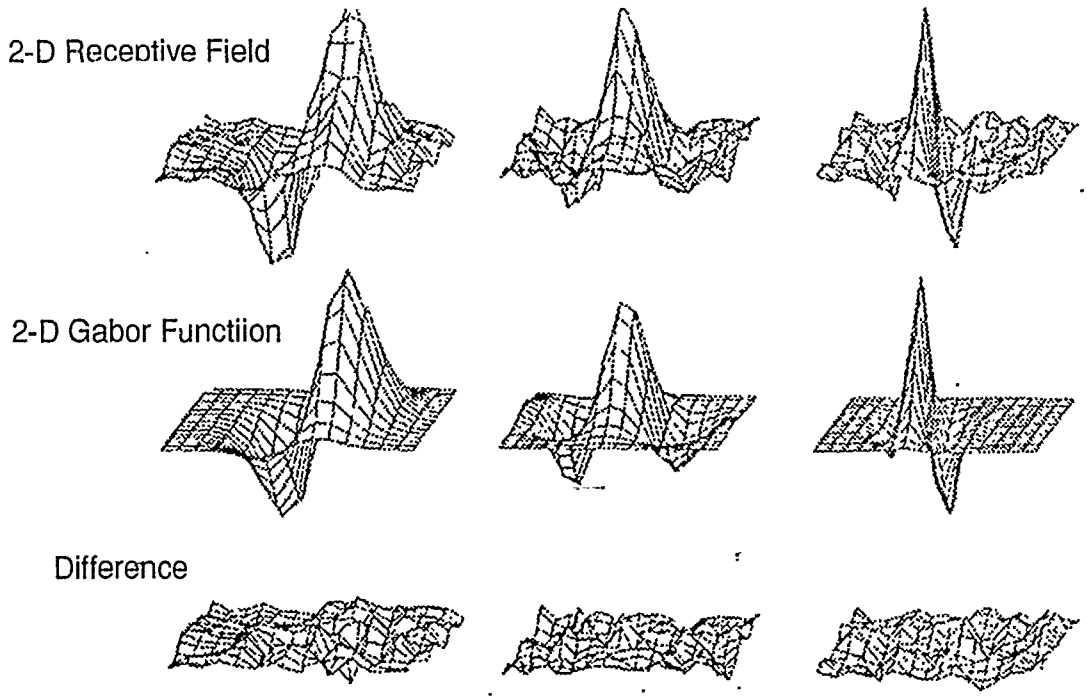

Figure 2. Comparison of 2-D Simple Cell Receptive Fields to Gabor Functions (5). The first row shows the measured two-dimensional response in simple cells of the cat visual cortex. The second row shows a set of "best fit" Gabor functions. The third row shows the residual error between the Gabor functions and the measured responses.

Bovik (2), Daugman (3), Lu (7), have used Gabor functions to process different types of imagery and have found the Gabor filters able to distinguish between textures in an image. Bovik stated,

> The Gabor implementation effectively unifies the solution of the conflicting problems of determining local textural structures (features, texture boundaries) and identifying the spatial extent of textures contributing significant spectral information ... (2:71)

2.2.2 Gabor Usage Daugman made use of a complete set of Gabor filters to use for image segmentation. The filters were based on a window size of 32 x 32 pixels, but could be generalized to any window size. In addition, the windows used were overlapping windows rather than independent windows as used in the previous effort of L'Homme. The general equation for these Gabor filters is (3:1176):

$$G(x,y) = \exp(-\pi\alpha^2[(x-mM)^2 + (y-nN)^2]) \cdot \exp(-2\pi j[r\frac{x}{M} + s\frac{y}{N}]) \quad (2)$$

Where: M, N - window dimensions

α^2 - Gaussian scale constant

m, n - integers indicating location of window

r, s - integer, increments of spatial frequency in the range $(-M/2-1, M/2)$ and $(-N/2-1, N/2)$ respectively

The Gaussian space constant, α, determines the roll-off of the Gaussian envelope. In this case α is set such that the value of the Gaussian envelope is $1/e$ at ± 9 pixels for a 32 pixel window size.

The frequency and rotation of the filters is determined by r, s, M and N as follows:

$$f_x = \frac{r}{M} \quad (3)$$

$$f_y = \frac{s}{N} \quad (4)$$

$$F = \sqrt{f_x^2 + f_y^2} \quad (5)$$

$$\Theta = \arctan(f_y/f_x) \quad (6)$$

Where: f_x = frequency in the x direction

f_y = frequency in the y direction

F = frequency of the filter

Θ = rotation of the filter

Daugman has also suggested a neural network to solve the problem of the non-orthogonality of Gabor functions when used as a basis set. A description of the network and its operation is given in Appendix A.

Several efforts at AFIT have concentrated in the area of Gabor segmentation and filtering of both Infrared (IR) and SAR imagery. Work accomplished by Kevin Ayer (1) involved Gabor filtering of Forward-Looking Infrared (FLIR) imagery. Ayer was able to successfully pick out targets, tanks, trucks, and jeeps, from the background. Ayer's work was done via computer while similar work was done optically by Christopher Veronin (15). With correct selection of filters, Veronin was able to pick out the targets from the background as well. In addition, Veronin was able to optically reproduce Bovik's texture discrimination. The latest work done with Gabor filters for segmenting SAR imagery was done by Albert L'Homme as mentioned in Section 1.2.

Since Gabor functions are biologically motivated and have proven useful to discriminate textures, these functions may be the way to front-end process imagery. Since the human brain has a throughput rate of only about 50 bits/sec (9), a large reduction in the amount of input data has to be accomplished in order for the brain to accomplish the task of recognition. Gabor processing of imagery does reduce the amount of data input and appears to provide a useful set of features for an image segmentation algorithm.

2.3 Radial Basis Functions

A Radial Basis Function network (RBF) is an ANN used to classify supervised data. Instead of each neuron firing as the result of a linear or nonlinear function of it's

input, they each have an associated Gaussian receptive field. The location and spread of each receptive field are controlled by two parameters, the mean and the standard deviation respectively. The are two methods available within Zahirniak's code (17:4-3) to set the mean of the Gaussian receptive field. The first method is to move the centers of the Gaussians to the locations of clusters of the data (class average) in the feature space and adjust the spread of the Gaussian to include co-located data of the same output class. The second method is to set the mean of the Gaussians to data points in the input data. In addition, Zahirniak's code (17:4-4) also allows several methods to calculate the spread, σ, of the Gaussian receptive field. The three methods are scale sigma according to class interference, scale sigma according to P–neighbor distance, and set sigma to a constant. For this research, centering the Gaussians at class averages and scaling sigma according to class interference will be used. An example of a two-layer RBF is shown in Figure 3. The nodes of the output layer compute a linear combination of outputs from the hidden layer nodes. This makes the output mapping for any single input pattern

$$y_m = \sum_{l=1}^{L} w_{lm} y_l$$

and the output from the hidden layer nodes, y_l is

$$y_l = e^{-[\sum_{k=1}^{K} \frac{(x_k - w_{kl})^2}{2\sigma_{kl}^2}]}$$

Where: $x_k = k^{th}$ component of the input pattern vector $\vec{x}$

$w_{kl} = k^{th}$ component of the weight vector

$\sigma_k =$ the spread in the k^{th} direction.

The mean of the Gaussian receptive field for a RBF node is determined by its input weights. Since the weights determine a vector in the feature space, this vector locates the mean of the Gaussian in the feature space. The variance of the Gaussian function controls how widely spread the function is; the larger the variance, the more widely spread out it is the function within the feature space.

Daniel Zahirniak used RBF's to identify a radar by its transmitting characteristics (17). In addition, he also described the use of a RBF network for pattern

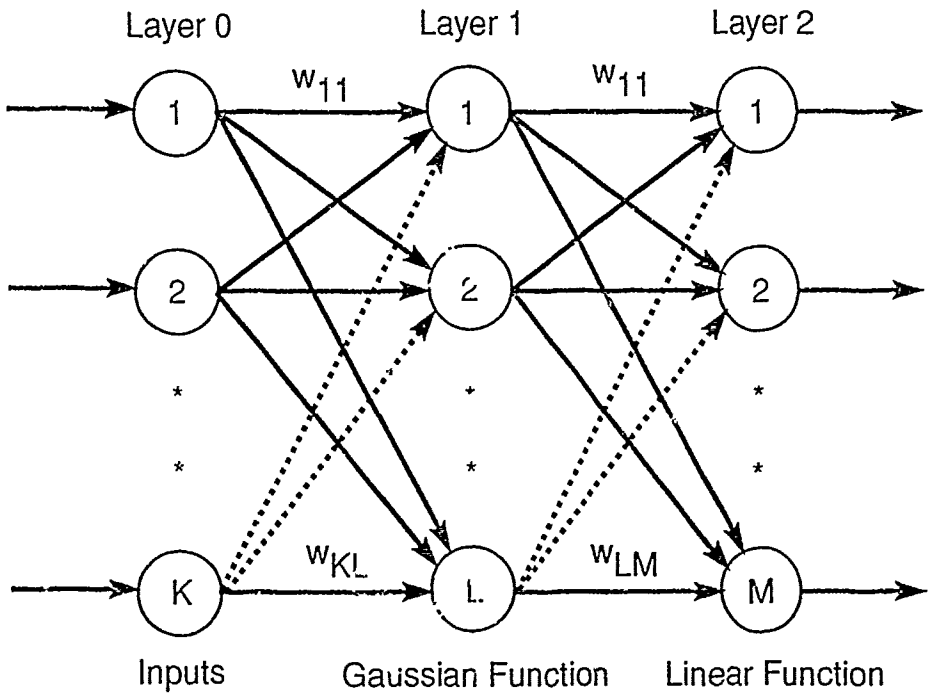

Figure 3. Two Layer Radial Basis Function (17:3-14)

recognition (18). L'Homme used a RBF to learn the features which best identify trees, grass, shadow, and culture. Work by Priddy (17) makes use of a RBF to segment Gabor filtered IR imagery. With such a large amount of experience and knowledge of RBFs at AFIT, it will be the ANN used for segmentation of the SAR imagery. Since there is no apriori knowledge of which Gabor coefficients match with a particular image feature, it is hoped that a RBF can be used to learn the correct combinations of coefficients.

2.4 Summary

In this chapter, the biological connection of the Gabor function and its advantages in texture discrimination for various types of imagery were discussed. In addition, the successful use of a RBF as a segmenter for a range of imagery in previous work was covered.

With these tools that have shown promise to aid in segmenting imagery, a more robust method for segmenting SAR imagery should be attainable.

III. Methodology

This chapter will cover the methodologies used to segment the high resolution SAR imagery. First, the preprocessing done to each SAR image will be shown. Second, will be a description of the sliding window and Gabor functions used. Third, the method of computing the optimal Gabor coefficients will be explained. Fourth, the implementation and training of a RBF to select the proper classification of the input data will be covered. Finally, image segmentation will be discussed.

3.1 SAR Image Preprocessing

Each SAR image is composed of four frames of dimension 512 x 2048 pixels resulting in an image size of 2048 x 2048 pixels. These frames contained low return areas on both the left and right side of the image due to roll-off of the main beam of the radar as can be seen in the dark areas in Figure 1. The radar was calibrated for the center of the main beam which corresponds to the center of the image. Thus, the center 1024 pixels from each frame were used for this research.

In an attempt to normalize the data from different missions, each reduced (512 x 1024) frame was Fourier transformed via an FFT, the DC component was set to zero, and the frame was inverse transformed.

3.2 Gabor Filter Generation

The initial set of Gabor filters used will be those based upon filters suggested by Daugman as mentioned in Section 2.2.2. These form a complete set of Gabor filters for a window size of 32 x 32 pixels. Using the same Equation 2, a complete set of Gabor filters for a window size of 16 x 16 pixels will be generated. In the case of a 32 pixel window size, this will result in 256 filters. In the other case of a 16 pixel window, the result will be 64 filters.

3.3 Gabor Coefficient Calculation

The Gabor coefficients for the SAR imagery will be computed by correlating a set of complex Gabor filters with the complex SAR image. The correlation will be performed in the space domain through the use of a sliding window. The window

size will be varied from a starting size of 32 x 32 pixels to a size of 16 x 16 pixels. The window will start in the upper left-hand corner with its center placed at the same location in the image as in the window. For example, for a window size of 32 x 32 pixels the starting location would be at pixel location (16,16) on the image. This also corresponds to the center of the window which is at (16,16). The window will the be moved across the image in steps of $\frac{1}{2}$ the window size both horizontally and vertically. At each location, the Gabor filter will be correlated with the image yielding a set of correlation coefficient.

3.4 Gabor Filter Selection

Two methods will be used to find the optimal set of Gabor filters for use in the training of the RBF. The first method suggested by Bovik (2:63) is to compute the power spectrum of the image and select the two highest peaks and select the Gabor filter associated with that location in the spatial frequency domain. The second method involves histogramming of the Gabor filter coefficients. Coefficients will be calculated for windows containing only a known region of trees, grass, and shadow. For each window the three filters having the largest magnitude coefficient will be tallied. After all the windows have been processed, the final counts will be histogrammed. The filters having the largest number of occurrences will then be used as the set of filters to process entire SAR images with.

3.5 RBF Training

The radial basis function to be used will be implemented using the neural network software developed by Zahirniak (17). The software allows the user to choose the entire architecture of the network and select from among several training methods for establishing the network weights.

The number of cluster centers will be varied in order to find the optimal number. The limitation to this RBF is that the number of training vectors is limited to 200. This limitation is imposed by the size of the matrix inversion routine implemented by Zahirniak.

All input data will be statistically normalized. This procedure involves calculating the mean and standard deviation for each feature of the input vector. The data for each feature is then normalized according to a Gaussian distribution as follows.

$$x_i^j = \frac{x_i^j - \mu_i}{\sigma_i}$$

Where: x_i^j = the i^{th} feature of the the j^{th} input vector

μ_i = the mean of the i^{th} feature

σ_i = the standard deviation of the i^{th} feature.

To optimize the training of the RBF, regions of images containing only trees, grass, shadow, or culture will be selected. The Gabor coefficients for each of these regions will be used as training vectors. Several sets of training data will be created and several variations of the RBF will be used. The initial variation of the RBF to be used will be initially placing a node (Gaussian receptive field centers) at the class average.

Once the "best" RBF has been selected, the remaining SAR imagery will be processed through the network and the resulting outputs will be used to segment the images for comparison to the hand segmented images.

3.6 Image Segmentation

Image segmentation will be accomplished by assigning a unique grayscale value to each of the classes of data (trees, shadow, grass, and culture). Reading in the output from the RBF, the image will be drawn by shading each pixel to the appropriate grayscale value. A copy of the original hand segmented image will be compared to the computer generated image to find any errors present.

3.7 Summary

In this chapter the methodology for segmenting SAR imagery was covered. Specifically the methods for calculating the Gabor coefficients, training and using a RBF for segmenting, and image segmenting were discussed.

IV. Results

This chapter will address the SAR imagery used, the results obtained from the two techniques used to find the optimal Gabor filters, training the RBF, the RBF segmentation of entire SAR frames, and image reconstruction.

4.1 SAR Imagery

The SAR imagery used for this research came from two different ADTS missions.

- Mission 85 Pass 5 Stockbridge, New York.
- Mission 98 Pass 3 Portage Lake, Maine.

Below is given a description and a log-scaled image of each frame used for this research.

4.1.1 M85F27 This image contains a large region of trees running through the center of the image with areas of field along either side of the trees. Note the small area of scattered trees in the bottom left hand corner and the shadow region to the right of the large center tree region.

4.1.2 M85F28 This image also contains a large region of trees through the center of the image with a small area of field to the left of the trees and an area of shadow to the right of the trees.

4.1.3 M85F30 This image contains mostly fields with a small line of trees running down the right side of the image. A small road runs down the left side of the image. In addition, a calibration array of eight corner reflectors is present at the center of the image.

4.1.4 M98F07 This frame contains nearly all trees with a small area of field in the upper left hand corner of the image.

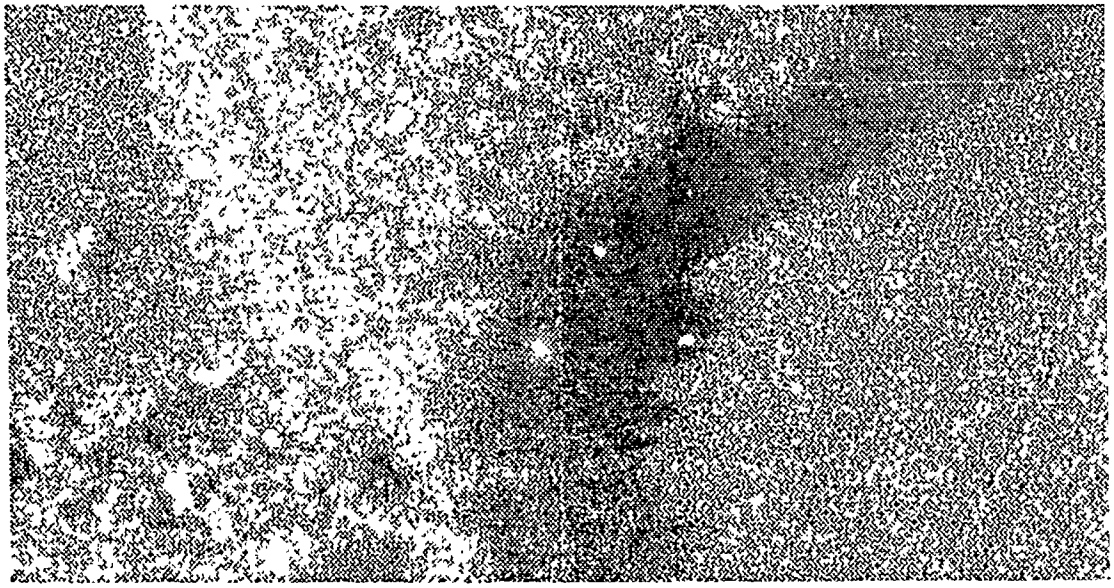

Figure 4. Mission 85, Frame 27, HH Polarization

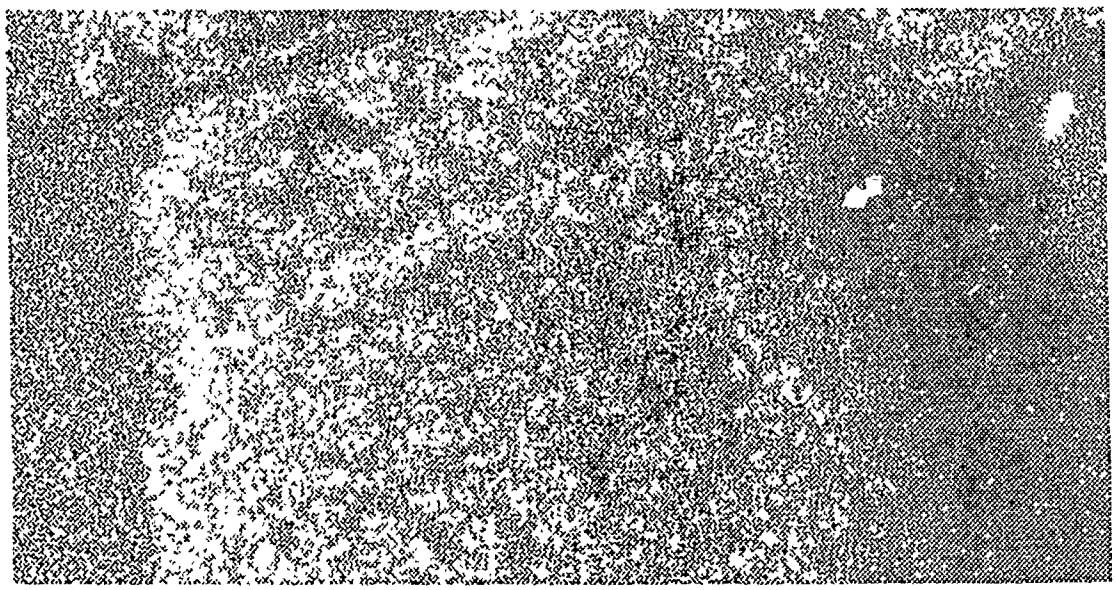

Figure 5. Mission 85, Frame 28, HH Polarization

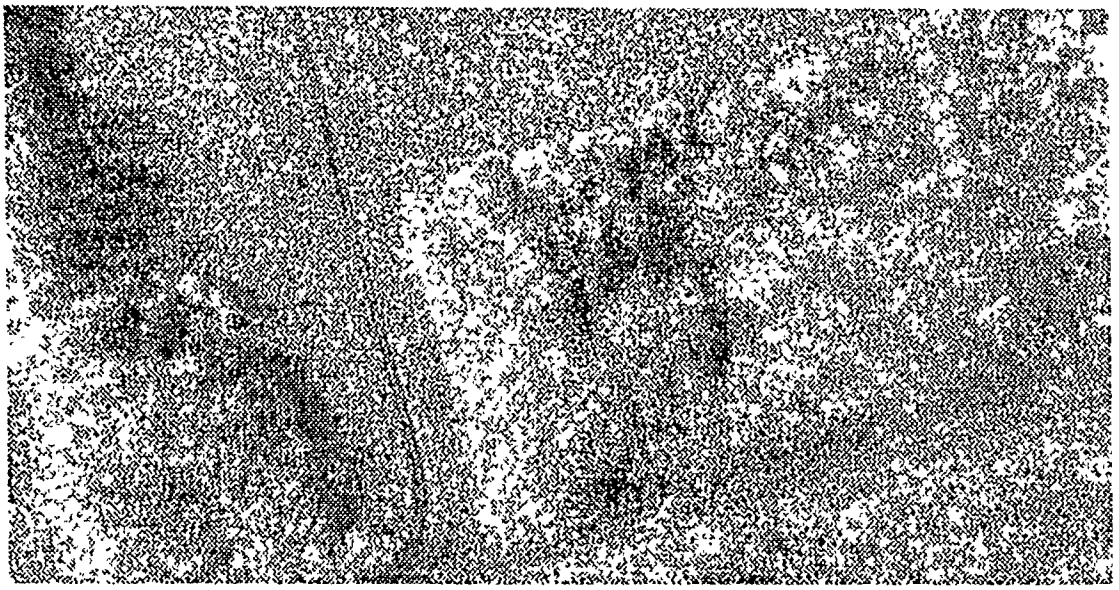

Figure 6. Mission 85, Frame 29, HH Polarization

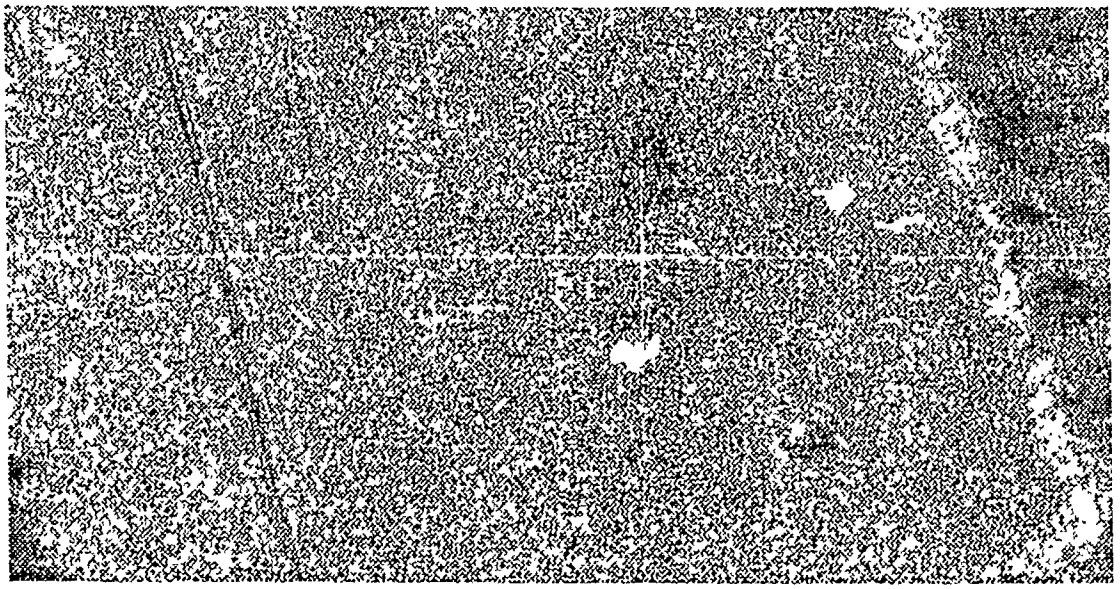

Figure 7. Mission 85, Frame 30, HH Polarization

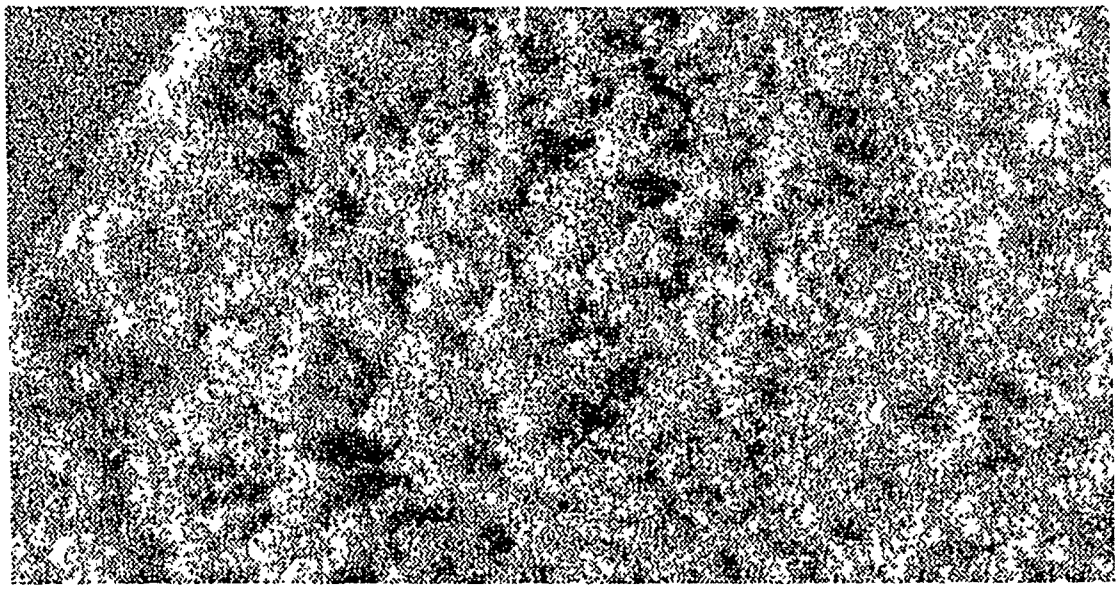

Figure 8. Mission 98, Frame 07, HH Polarization

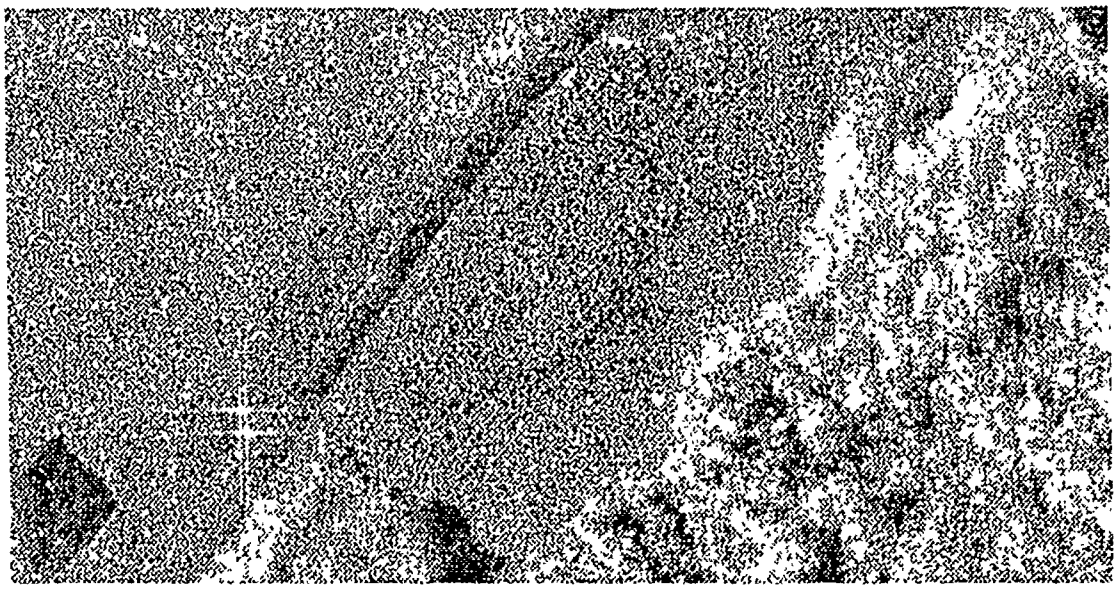

Figure 9. Mission 98, Frame 08, HH Polarization

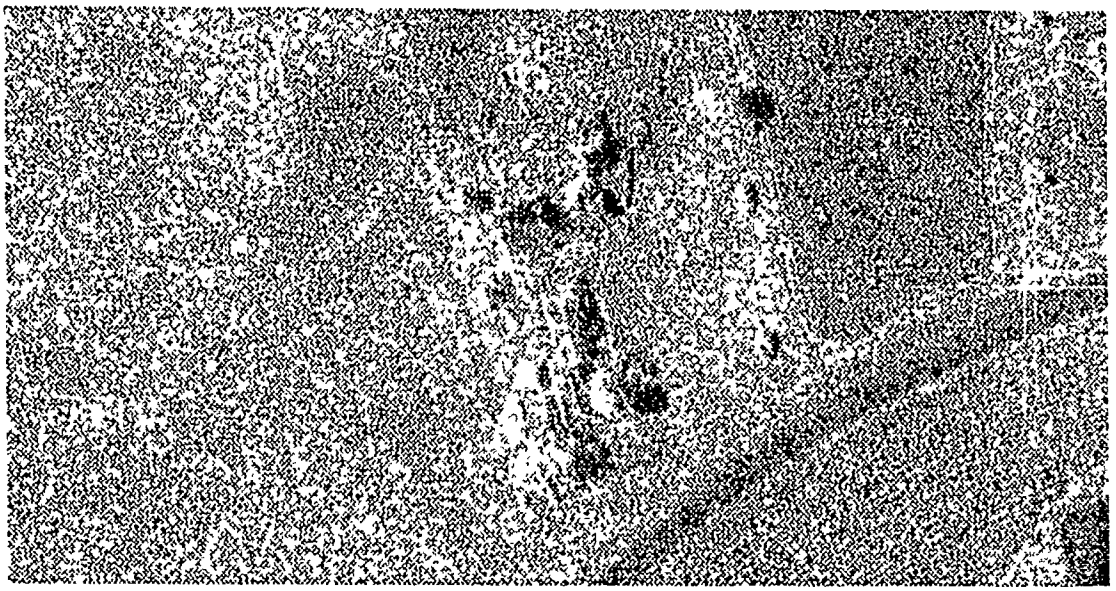

Figure 10. Mission 98, Frame 09, HH Polarization

4.1.5 M98F08 This image contains regions of tree, field, water, and road. There is a small pond located in the lower left hand corner of the image with a road running diagonally through the image from the same corner. The right hand side of the image contains a region of trees. In addition, there are two corner reflectors located just off the left side of the road.

4.1.6 M98F09 This is the only image that contains any cultural features. The center of the image contains several farm buildings, a house, sheds, a garage, and several vehicles. There is a road and a small pond located in the lower right hand corner. The rest of the image is composed of fields.

4.1.7 M98F10 This image contains another array of five corner reflectors located in what appears to be a mowed or plowed field. Notice the difference in contrast between the field where the corner reflectors are located and the surrounding field areas.

4.2 Correlation Coefficient Calculation

The correlation of the complex SAR image with the set of complex Gabor filters was performed by the program correlate.c. This program performed a correlation by

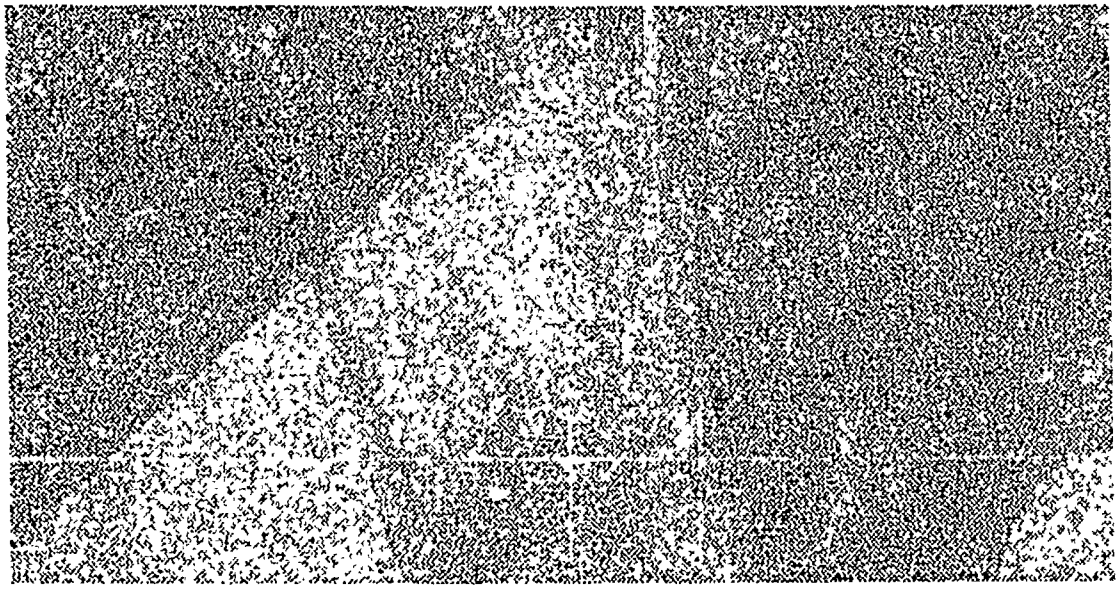

Figure 11. Mission 98, Frame 10, HH Polarization

moving a sliding window of the size specified by the user over the image. Each Gabor filter was then multiplied point-by-point with the image at that window location. The result was a set complex coefficients representing the value of the correlation of each filter with the image at that window location.

4.3 Gabor Filter Selection Using Fourier Technique

Several sections of trees, grass, shadow, and culture were clipped from frames in Mission 85 and Mission 98. Within the Khoros image processing software (14), a Fast Fourier Transform (FFT) and the power spectrum were calculated for 32 x 32 and 16 x 16 pixel subsections of these sections. Using the thresholding routine within Khoros, the power spectrum was thresholded until only two components remained. Examples of the power spectrums for trees, grass, and shadows can be found in figures 12, 13, and 14 respectively. It should be noted that the magnitude of the power spectrum for shadow regions is extremely low by comparison that of trees and grass. This trend was observed across all data analyzed. After computing the dominant spectrum components, it was discovered that both trees and grass had nearly the same dominant spectrum components and that the spectrums were

Table 1. Table showing the dominant power spectral components for 32 x 32 and 16 x 16 pixel regions of trees, grass and shadow. Total region sizes used were 256x256 pixels for trees and grass and 128x128 pixels for shadow. The components shown are common to both sets of mission data.

class	freq (cyc/window)	Rot (degrees)			
trees	1.000	0	90	180	-90
	1.414	45	180	-45	-135
	2.236	116	-63		
grass	1.000	0	90	180	-90
	1.414	45	180	-45	-135
	2.236	116	-63		

scattered at low frequencies. A partial list of the components found is shown in Table 1.

4.4 Histogramming Technique

Initially the sections of trees, grass, and shadows clipped from the original images were correlated with a set of 256 filters of window size 32 x 32 pixels. The correlation was performed by the correlate.c program which is listed in Appendix C.1. Another program, maxval.c read in the file containing the coefficient values and calculated for each window the three filters with the largest coefficients. A running count of the three largest filters was kept and the results stored in a file.

The results of histogramming the coefficients for a 32 x 32 window size for regions of trees, grass, and shadow are shown in Figures 15–17, respectively. Even though the filters used extended to frequencies above twelve cycles/window, there was no place in the in any of the regions of trees, grass, or shadow where frequencies above twelve cycles/window dominated. In each case, the histogram showed that the dominant frequencies were those below six cycles/window.

At this point, the equation which generated the Gabor filters was modified to allow a scaling of the frequencies used. Recall that in Equation 5 the terms $(\frac{r}{M}, \frac{s}{N})$ controlled the frequency of a filter. The new modification changed these terms to $(\frac{r}{kM}, \frac{s}{kN})$ where k is an integer scaling factor. By increasing the scale factor and allowing r, s, M, N to remain the same, the frequency of any filter would be reduced.

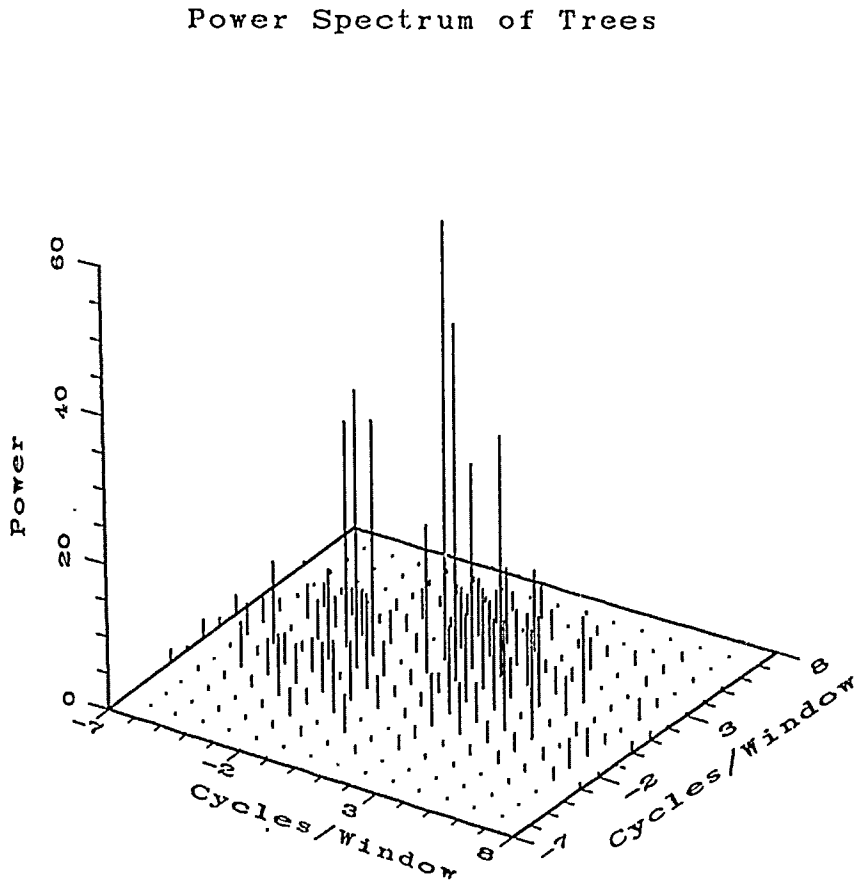

Figure 12. Typical Power Spectrum of 16 x 16 Window of Tree Region

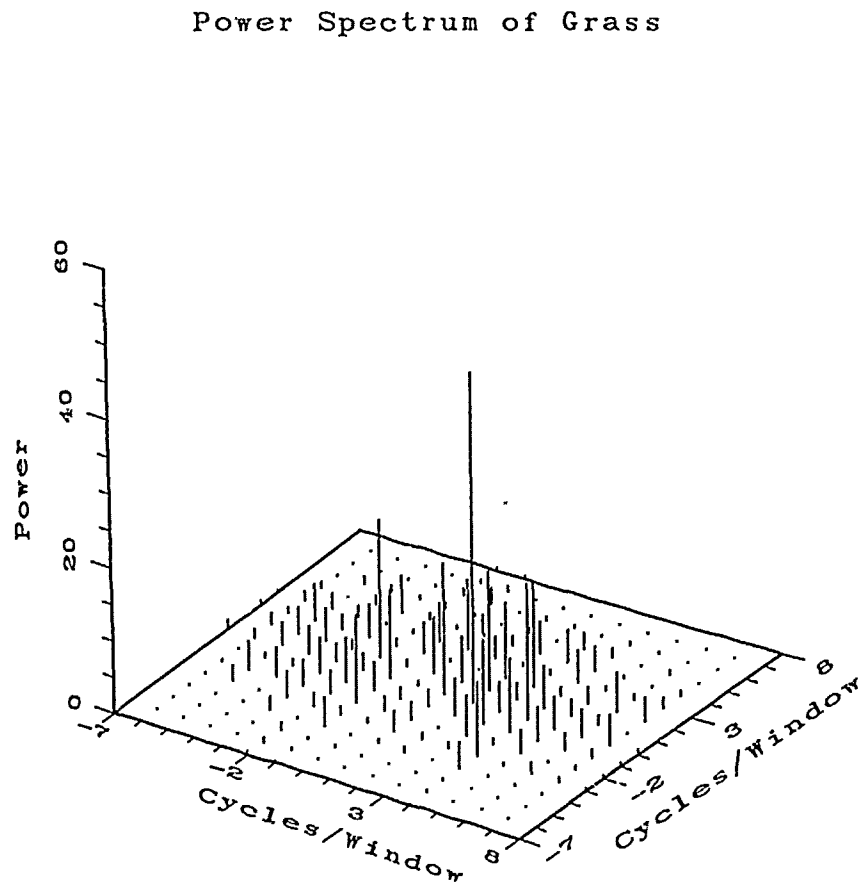

Figure 13. Typical Power Spectrum of 16 x 16 Window of Grass Region

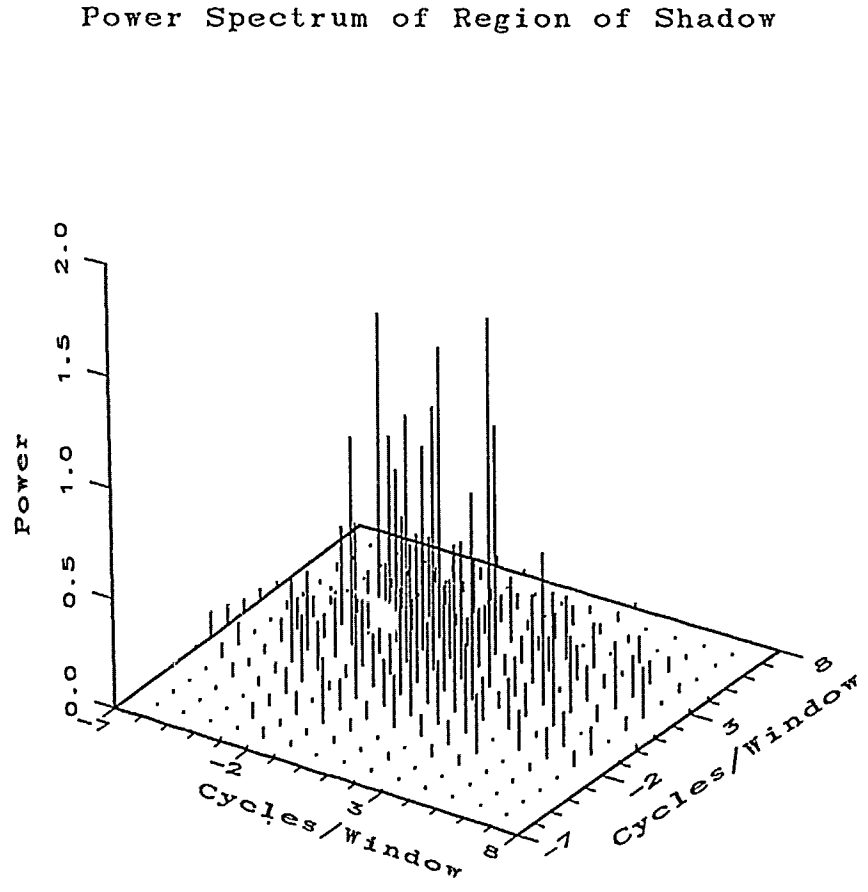

Figure 14. Typical Power Spectrum of 16 x 16 Window of Shadow Region

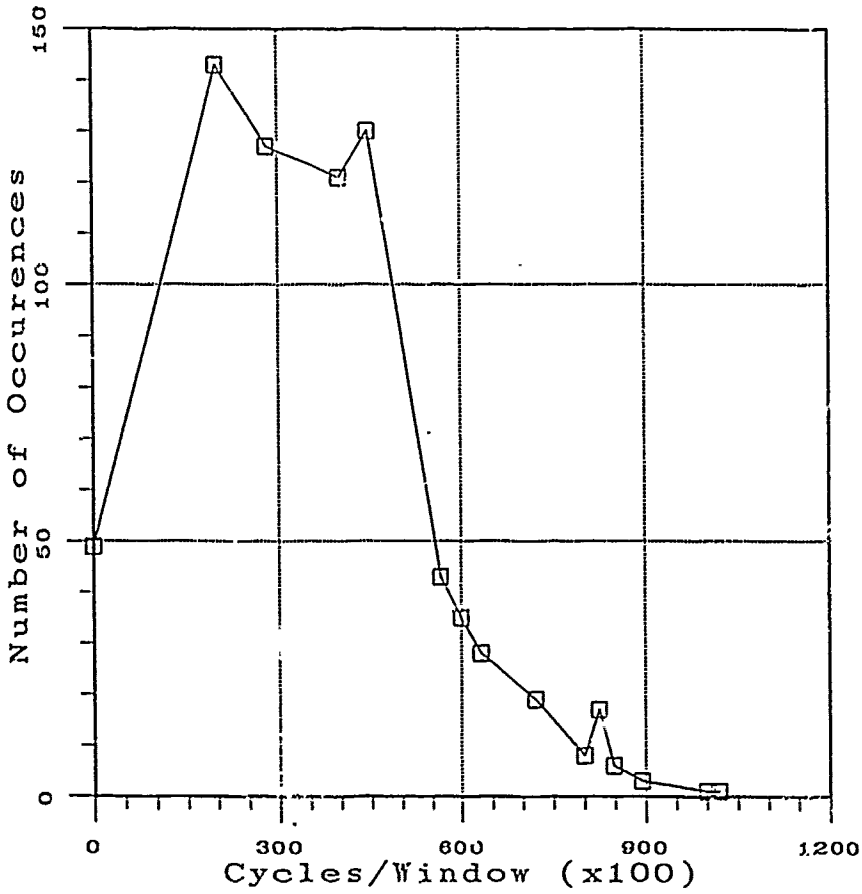

Figure 15. Histogram showing the number of times a filter, with the given frequency, was among the three dominant filters for a 32 x 32 pixel window in a 256 x 256 pixel region of trees.

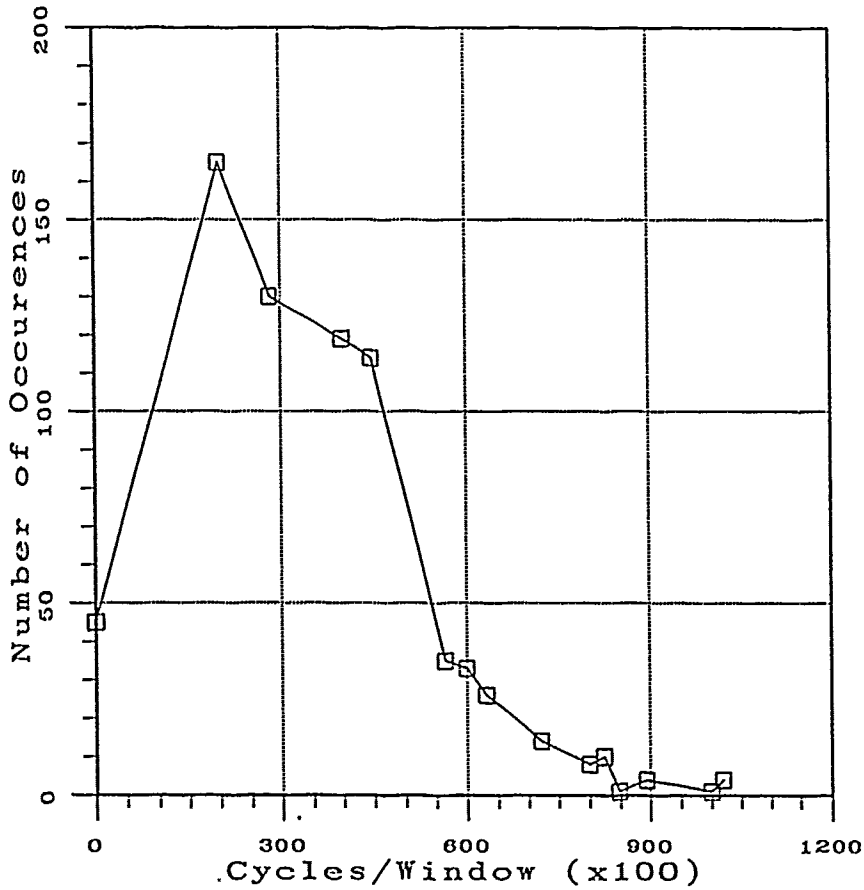

Figure 16. Histogram showing the number of times a filter, with the given frequency, was among the three dominant filters for a 32 x 32 pixel window in a 256 x 256 Pixel region of grass

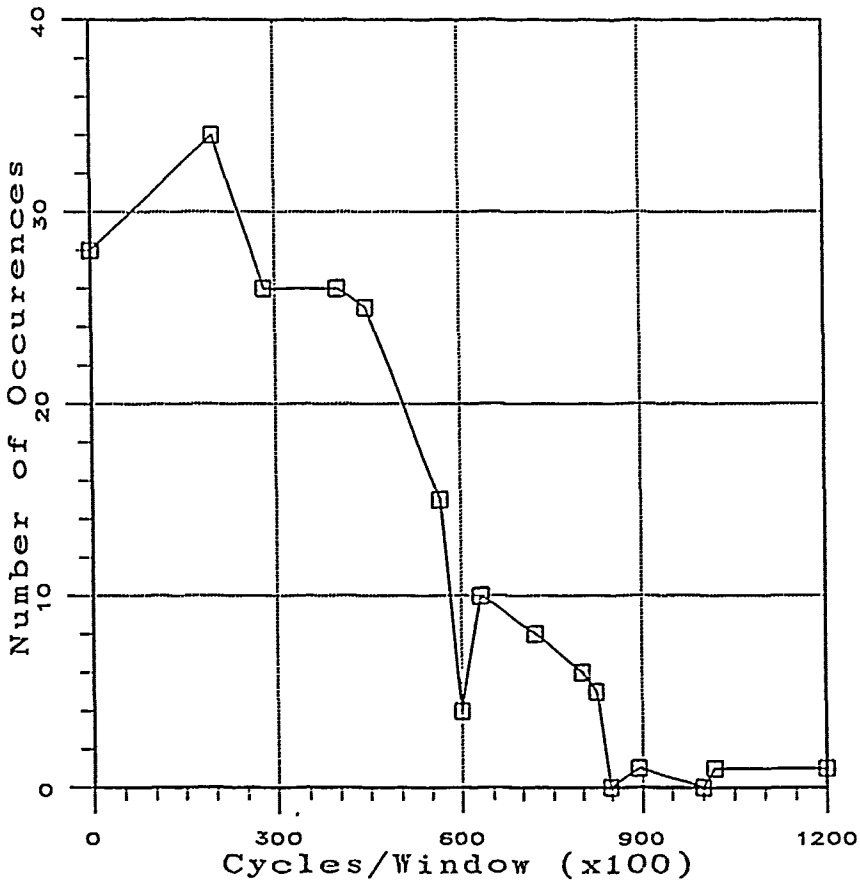

Figure 17. Histogram Showing the number of times a filter, with the given frequency, was among the three dominant filters for a 32 x 32 pixel window in a 128 x 128 pixel region of shadow

Table 2. Selected Filters for Image Processing

Freq (cyc/window)	Rotation (deg)
0.000	0
1.000	0 90 180 -90
1.414	-45 -135
2.000	0 180
2.236	-27 -153
3.000	0

The scaling factor used was four. Thus, the maximum frequency for the 256 Gabor filters was 5.65 cycles/window rather than 22.63 cycles/window.

The new set of filters was correlated with the sections of trees, grass, and shadow. A new histogram of the coefficients was generated. The new histograms, shown in Figure 18 and

Figure 19, and Figure 20 revealed that there were no dominant filters to be found at this window size and resolution. Therefore, the window size was reduced to 16 x 16 pixels in order to obtain better resolution.

Using the insight gained from the Fourier analysis and the 32 pixel filters, the new filters were generated by changing M and N in equation 2 to 16 and a scale factor of two was used in order to keep the maximum frequency less than six cycles/window. Results from the histogram of the coefficients from this set of filters showed dominant frequencies. The histograms for this set of filters are shown in Figure 21, Figure 22, and Figure 23.

From the initial set of 64 filters, 12 filters were selected to use as inputs to the RBF. The filters used are listed below in Table 2. These filters corresponded to the filters which responded best to each of the region types.

4.5 RBF Training

The options used with the RBF were to center the nodes at class averages and to train the weights by matrix inversion. The interference threshold for the nodes was set at 0.4. This implies that if a node of another class responded with a value larger than 0.4 that the RBF needs to continue to train.

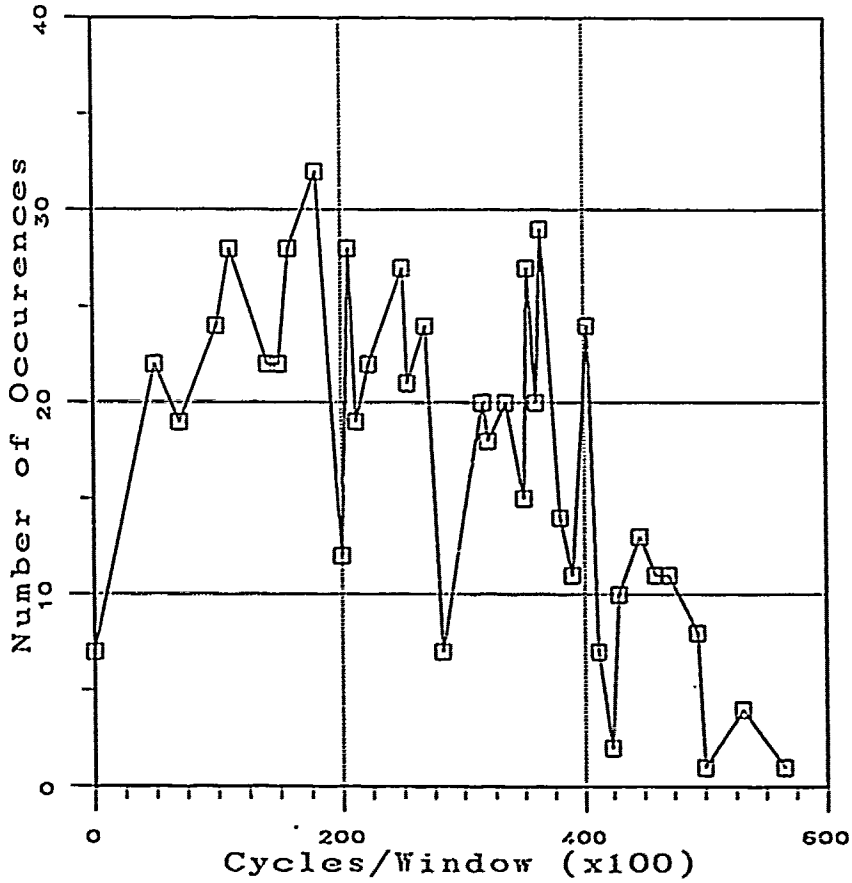

Figure 18. Histogram showing the number of times a filter, with the given frequency, was among the three dominant filters for a 32 x 32 pixel window in a 256 x 256 pixel region of trees. Scale factor of four applied to original filter frequencies.

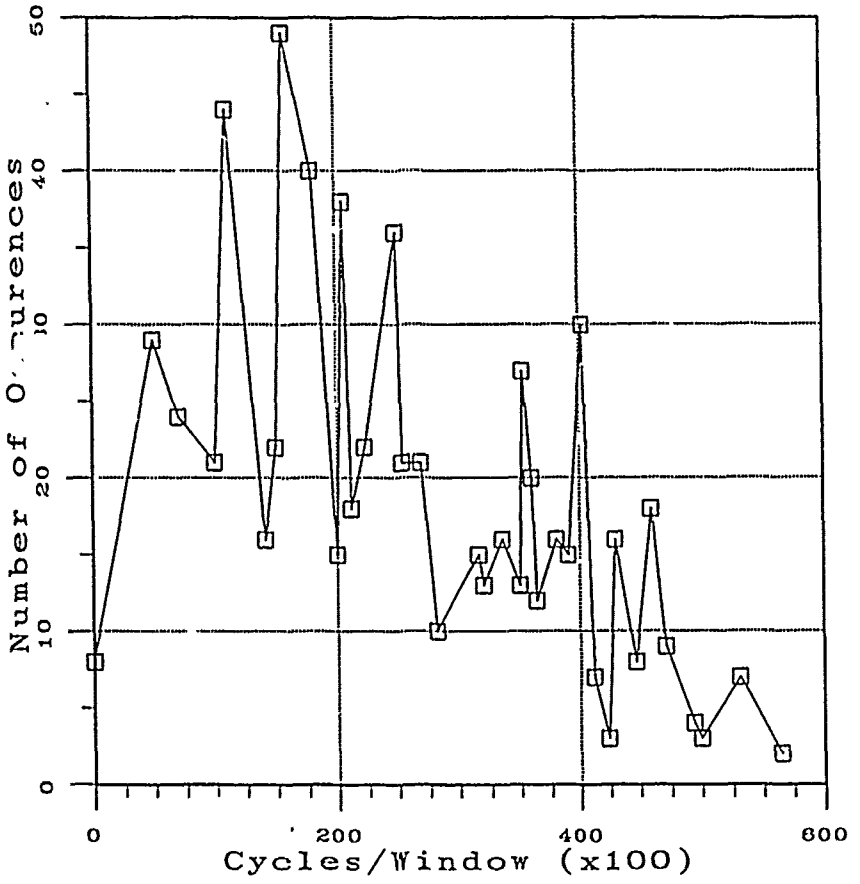

Figure 19. Histogram showing the number of times a filter, with the given frequency, was among the three dominant filters for a 32 x 32 pixel window in a 256 x 256 Pixel region of grass. Scale factor of four applied to original filter frequencies

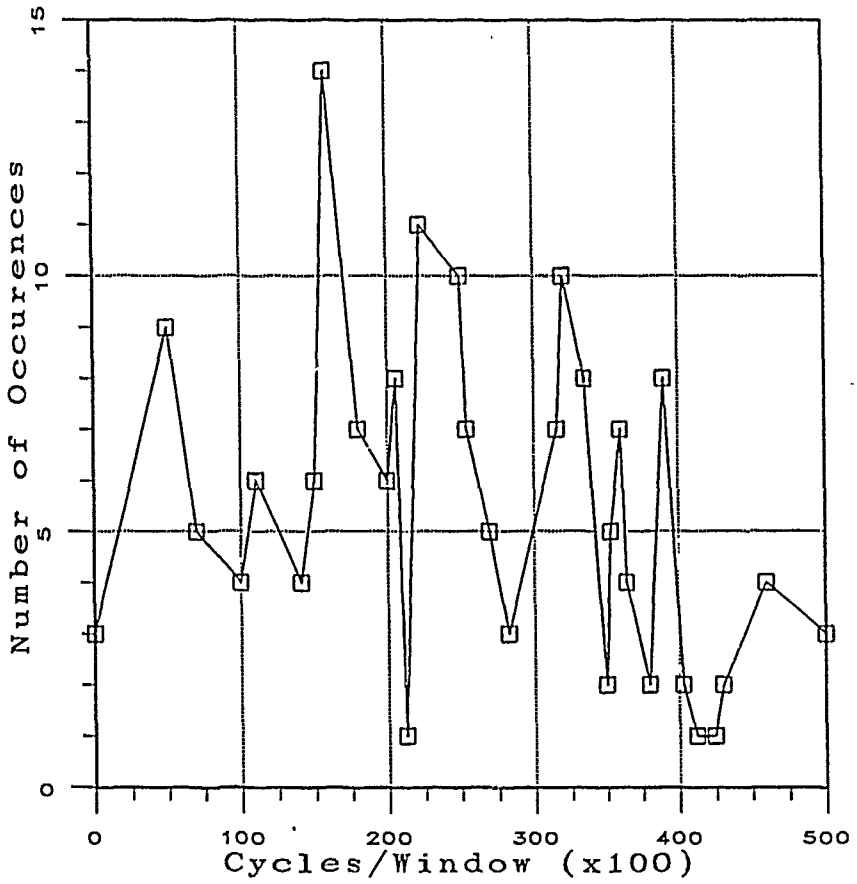

Figure 20. Histogram Showing the number of times a filter, with the given frequency, was among the three dominant filters for a 32 x 32 pixel window in a 128 x 128 pixel region of shadow. Scale factor of four applied to original filter frequencies.

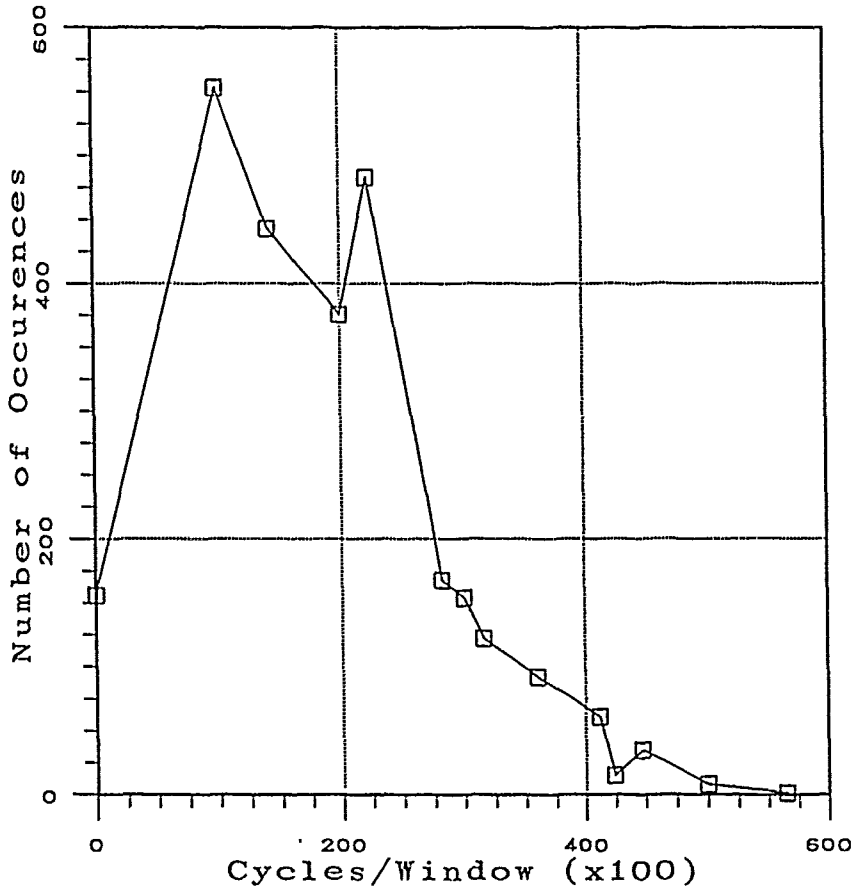

Figure 21. Histogram showing the number of times a filter, with the given frequency, was among the three dominant filters for a 16 x 16 pixel window in a 256 x 256 pixel region of trees. Scale factor of two applied to original filter frequencies.

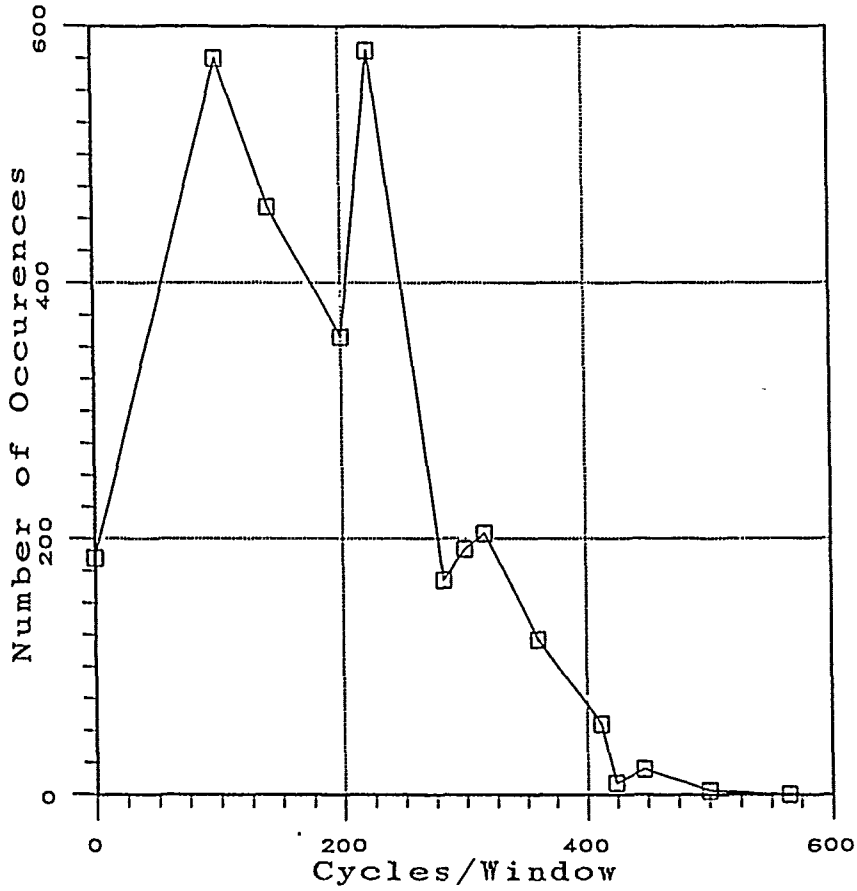

Figure 22. Histogram showing the number of times a filter, with the given frequency, was among the three dominant filters for a 16 x 16 pixel window in a 256 x 256 Pixel region of grass. Scale factor of two applied to original filter frequencies

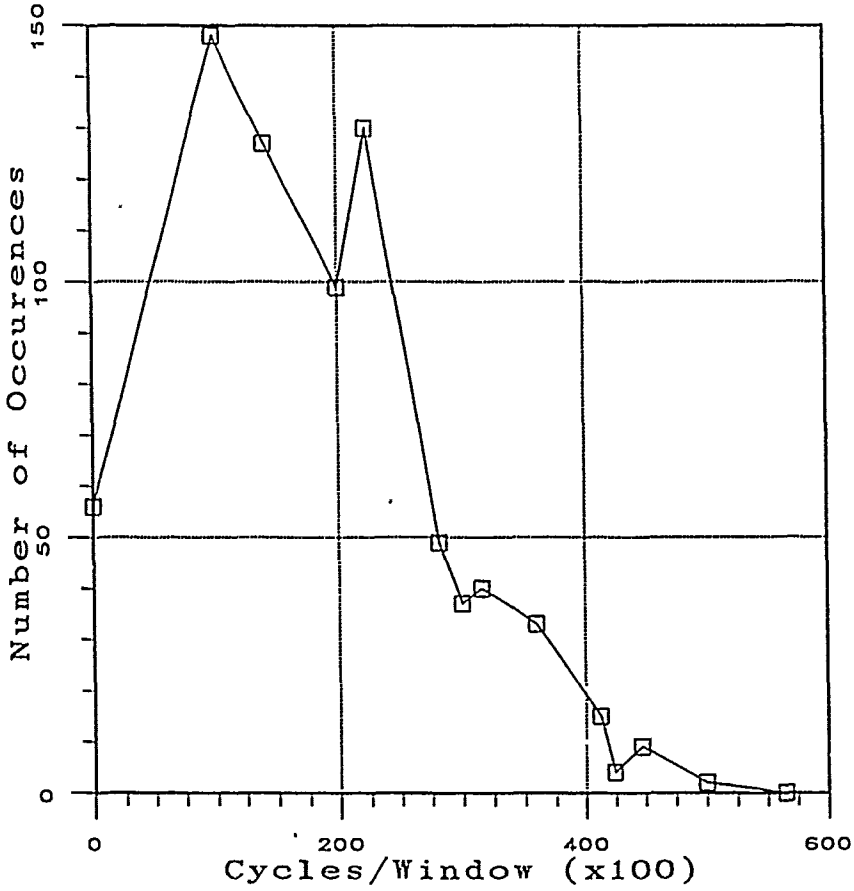

Figure 23. Histogram Showing the number of times a filter, with the given frequency, was among the three dominant filters for a 16 x 16 pixel window in a 128 x 128 pixel region of shadow. Scale factor of two applied to original filter frequencies.

The RBF was trained using cross-validation technique. In other words, initially data from Mission 85 was used as training data while Mission 98 data was used as test data. Next, the process was reversed and Mission 98 data was used for training and Mission 85 data was used for testing. In each case, 50 vectors for every class of data were used for both training and testing. The training and testing vectors were sampled from a large number of vectors available. Specifically, there were 2048 vectors for trees and grass regions and 496 vectors for shadow regions. The program pickval.c allowed sampling of these data files starting at any location and sampling at any integer interval. The training and test data were statistically normalized using the program norm.c. The normalization process was to aid the RBF in generalizing.

4.6 Training Results

The RBF was trained and tested 50 times using various combinations of training and test data. In every case, the RBF was able to achieve 97 to 100 percent accuracy classifying the training data correctly. However, depending upon the test data presented, the classification accuracy of the test data was as high as 93 percent and as low as 65 percent. Average accuracy for test data classification was 81 percent.

4.7 Segmentation Results

The first test image was Mission 98, Pass 3, Frame 07 (M98P3F07) shown in Figure 8. The hand segmented version of this same frame is shown in Figure 24. The hand segmented version showed two regions only in the image, grass in the upper lefthand corner and trees throughout the rest of the image. The RBF segmented version of this frame is shown in Figure 25. It appeared that the RBF segmented version was finding more detailed features than originally expected.

Other RBF segmented images can be found in Appendix B. Overall, it was not possible to compare the RBF segmented images to the hand segmented images. In each case, the RBF segmentation was more detailed than the hand segmented version. The hand segmented versions of the frames were generated based upon the assumption that the best that could be accomplished was to find macro features in the imagery and not detailed features.

It is apparent from the RBF segmented imagery that the transition regions between trees and shadows is often misclassified as regions of grass. This apparent

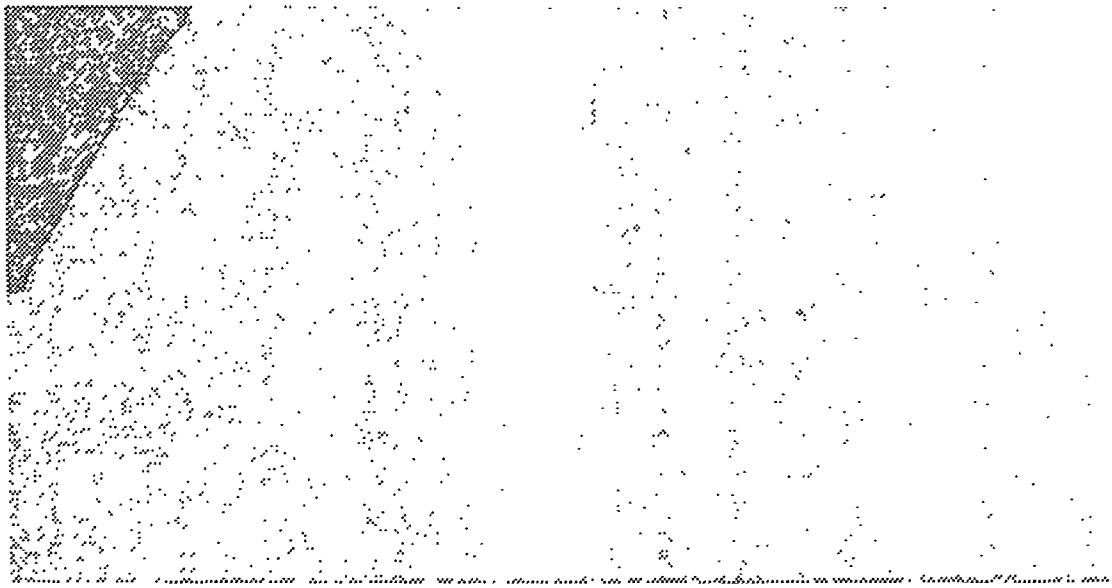

Figure 24. Mission 98, Frame 07 Hand Segmented

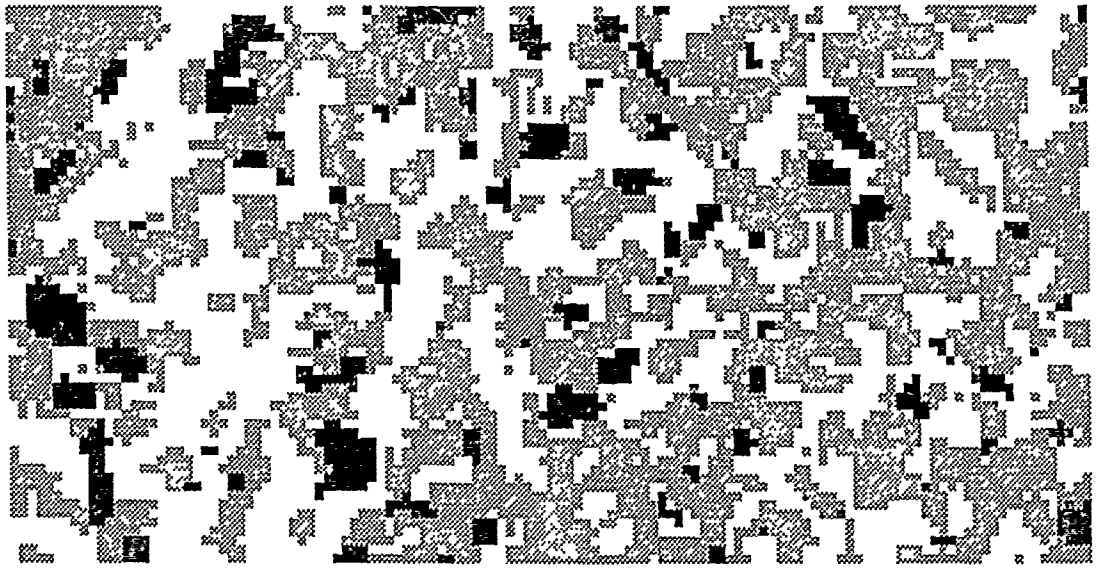

Figure 25. Mission 98, Frame 07 RBF Segmented

weakness could be explained by the fact that all of the filters used to process the SAR imagery were of low frequency. The lack of high frequency information makes it difficult to find edges.

It appeared that the discrepancy in the accuracy of the ability of the RBF to classify the test data correctly could be explained by the fact that the regions of trees in the Mission 98 data contained areas of shadow. In addition, after examining the previous work of L'Homme, it was discovered that the data from Mission 98 contained higher radar return values for areas of trees and grass than did Mission 85. This could also be a source of error.

One other segmented image of particular interest is Mission 98, Frame 10 shown in Figure 26. This image contains regions of grass with an array of corner reflectors in it. When the photograph of this area was examined, the field in which the corner reflectors were placed turned out to be a plowed field. After segmentation, the plowed field was segmented as a region of shadow rather than a region of grass. The grayscales used in this segmented image are different from all others shown in this research. The correct interpretation of the grayscales is the darker grey regions are grass, light grey regions are shadow, black regions are trees, and the white regions are corner reflectors. It should be noted that even though the RBF network was not trained to identify corner reflectors, the output of the network was all zeros when a region containing corner reflectors was being processed. This indicated that the input vectors for corner reflectors were very distant in the feature space from any of the class nodes.

In all other images with corner reflectors in them, the RBF output file was modified. The zeros that marked the location of corner reflectors were changed to match the grayscale of the type of region surrounding the reflector.

4.8 Median Filtering

In order to "clean up" the segmented imagery, some images were median filtered. As can be seen in Figure 25, there are small patches of grass regions within the region of trees. Each of these small patches are of size 8 x 8 pixels and correspond to a single output value of the RBF network. In order to remove these small patches a 5 x 5 pixel median filter was used on the RBF network output file. The median filtering was performed within the Khoros image processing software. An example

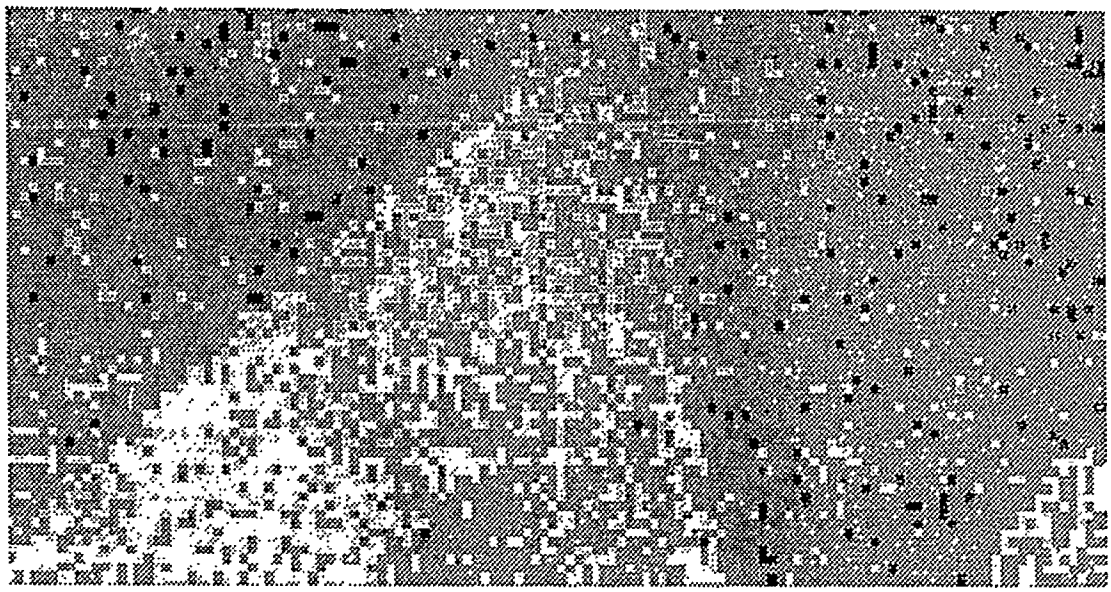

Figure 26. Mission 98, Frame 10 RBF Segmented

of the resulting image is shown in Figure 27 using the previous image from Misssion 98, Frame 07 as shown in Figure 25. Any other segmented image that was median filtered is noted in the figure caption.

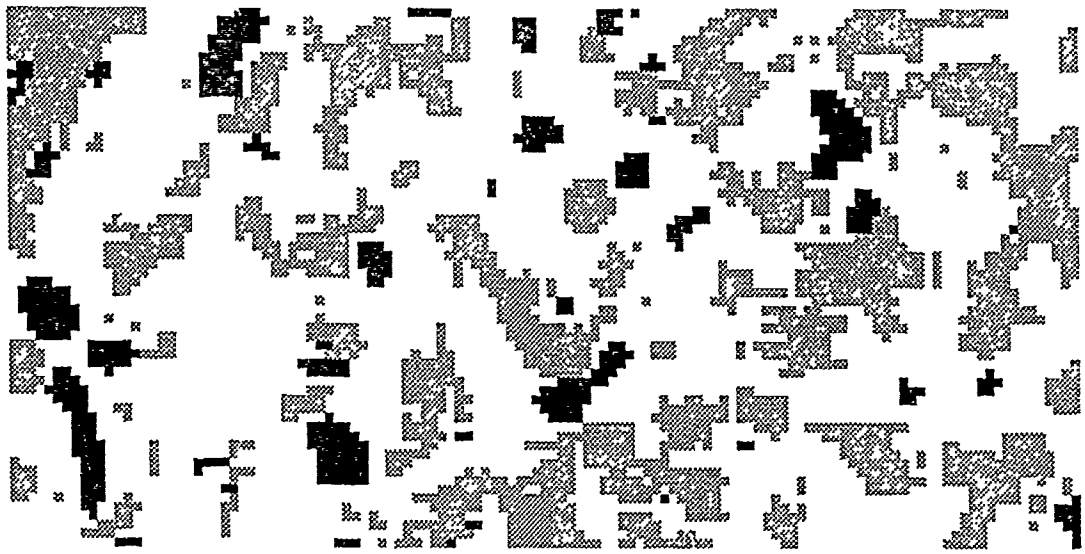

Figure 27. Mission 98, Frame 07 RBF Segmented, 5 X 5 Median Filter

V. Conclusions and Recommendations

5.1 Introduction

This research investigated the use of Gabor filters to segment high resolution SAR imagery. The aim was to find a set of Gabor filters which could be used to locate regions of trees, grass, and shadow using the complex representation of the SAR image. In particular, the following questions were answered during this work.

- Can the Fourier technique suggested by Bovik help to select the correct combination of filters to use for segmenting regions of trees, grass, and shadow?
- Can Histogramming techniques perform the same task?
- Can a RBF network learn the correct combination of filters to generalize to a full SAR image?

The Fourier technique showed that naturally occurring regions, in these SAR images, responded best to frequencies below 4 cycles per window (16 and 32 pixels). In addition, this analysis also showed that the phase of naturally occurring regions is widely scattered. In general, however, this technique did not work to isolate the correct Gabor filters to use. The cases which Bovik used were images with only one or two dominant textures in them. It appears that this technique is not useful for cases involving regions with more than two easily seen textures in them.

Histogramming was able to isolate a set of Gabor filters to use once the window size was reduced to 16 pixels. The Gabor filters isolated were all of low frequency (less than 4 cycles per window). Essentially, this technique found the bandpasses of frequencies that regions of trees, grass, and shadows responded to best.

The RBF network achieved testing accuracies as high as 93 percent and averaged 81 percent correct. However, due to the problem of normalization, the RBF network was not able to generalize well to entire images.

5.2 Further Research

Below are listed suggestions for further research.

- To allow consistent scaling across images, use another type of normalization. One suggested method would be a local normalization within each window.
- Find another technique for selection of the Gabor filters.
- Continue to use overlapping windows of other sizes such as 8 by 8 or 4 by 4. The overlapping windows appear to be able to find greater detail in the SAR imagery than do independent windows.
- Continue to use the RBF network. If the normalization problem is solved, the RBF should provide good segmentation across images.
- Try adding other polarizations to the Gabor processing as features for the RBF network.

Appendix A. *The Gabor Representation*

The only difficulty in working with the Gabor functions is that while the functions do form a complete basis set onto which to map an image, they are not orthogonal. This implies that when attempting to find the set of coefficients that appropriately map the image onto the Gabor functions, the inner product terms $[g_i(x,y) \cdot g_j^*(x,y)]$ between filters are non-zero. Recent literature by Daugman (3) has shown that the efficient way to find the Gabor representation of an image is through use of an artificial neural net (ANN). The ANN used contained three layers. The first layer was made up of a set of fixed weights assigned as the value of the Gabor functions. The second layers contains a set of adjustable weights that are used to multiply the output of the neurons in the first layer. The third layer is again made up of a set of fixed weights assigned as the value of the Gabor functions. The layer of adjustable weights, upon completion of learning, were the complex coefficients that represented the projection of the image onto the Gabor functions. These coefficients are the optimal coefficients in the sense of the minimal mean-squared-error. A more detailed proof of this mean- squared-error minimization is given below.

Computing of the Gabor coefficients will be accomplished by implementing the ANN suggested by Daugman (3) and shown below in Figure 28.

Using the Gabor functions as the set of basis functions to represent an image poses a difficult problem. Although the Gabor functions comprise a complete set of functions, that is, they can represent any function, they are not orthogonal. This means that the solution for the coefficients, a_{ij}, cannot be computed by the simple integration of the original image with the complex conjugate of the ith basis function $\phi_i(x,y)$. Starting from an image $I(x,y)$ represented as a sum of projection coefficients times a set of basis functions, each image pixel is computed as follows:

$$H(x,y) = \sum_{i=1}^{n} a_i \phi_i(x,y)$$

$$a_i = \int_{-\infty}^{\infty} I(x,y)\phi_i^*(x,y)\,dxdy - \sum_{j=1\ j\neq i}^{n} \int_{-\infty}^{\infty} I(x,y)\phi_i^*(x,y)\phi_j(x,y)\,dxdy$$

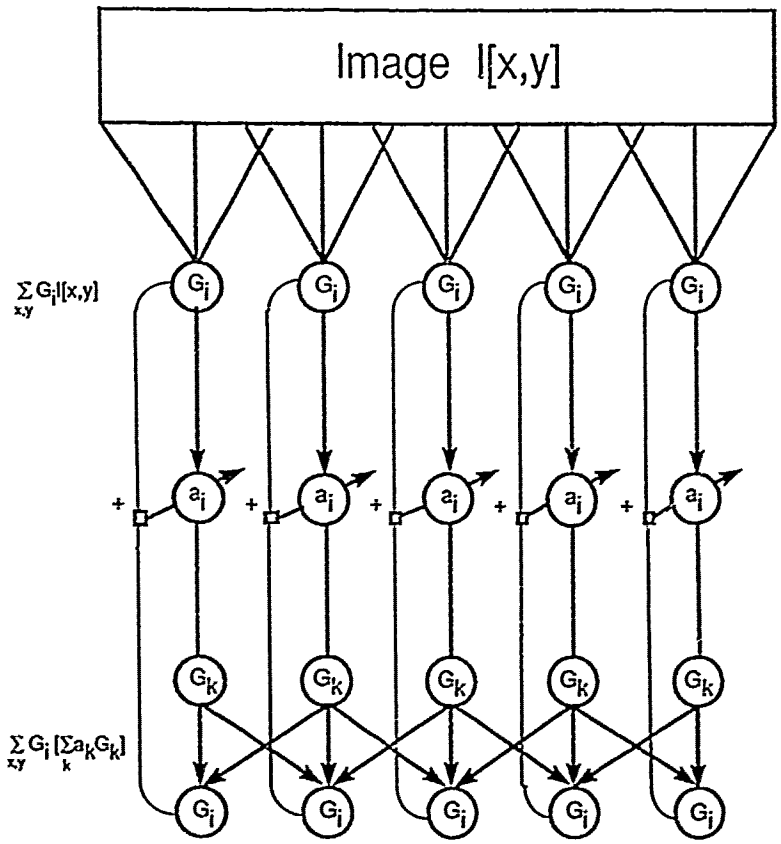

Figure 28. Three Layer Net for Finding Optimal Coefficients(3:1171)

where: $I(x,y)$ - original image

$H(x,y)$ - reconstructed image

a_n - expansion coefficients

$\phi_n(x,y)$ - basis function

Now had the original basis functions been orthogonal, all the terms in the solution for the a_i coefficient containing the form $\phi_i^*(x,y)\phi_j(x,y)$ would be equal to zero. But, because the set of Gabor functions is not orthogonal and because we only wish to use a subset of the Gabor functions, the best that can be accomplished is an optimal fit. The particular optimization criterion used by Daugman is that of minimizing the squared norm of the difference vector:

$$E = \| I(x,y) - H(x,y) \|^2$$

where: $H(x,y) = \sum_{i=1}^{n} a_i \phi_i[x,y]$ and

$\phi_i[x,y]$ are the elementary Gabor functions.

In order to minimize the norm, the partial derivative with respect to each of the coefficients must equal zero. Substituting for H and computing the derivative yields:

$$\frac{\partial E}{\partial a_i} = -2\sum_{x,y}[I(x,y)\phi(x,y)] + 2\sum_{x,y}[\sum_{k=1}^{n} a_k\phi_k(x,y)] = 0$$

The result is a set of n equations of the form:

$$\sum_{x,y} I(x,y)\phi_i(x,y) = \sum_{x,y}[\acute{o}_i(x,y)(\sum_{k=1}^{n} a_k\phi_k(x,y))]$$

Daugman points out that these equations could be solved with matrix manipulation, but the number of calculations necessary would be immense. He also points out that the error surface is quadratic in each a_i. Since neural nets can be used to minimize an error function, he proposes his net as a solution to this problem.

The net sets up the error surface as the squared norm of the difference between the original image and the current approximation. He then uses gradient descent to find the minimum of the error surface. Examining Figure 28, the first layer of the net yields the product of each of the elementary Gabor functions $\phi_i[x, y]$ with the original image (i.e. the Gabored image). The third layer of the net yields the inner product of each elementary function with the linear combination of all the functions (i.e. the reconstruction from the Gabor coefficients). Finally, the second layer is the layer used to calculate the coefficients. Initially, the second layer is set equal to the product of each Gabor function with the image. Then, by using gradient descent, the coefficients of the second layer are adjusted according to the following relationship:

$$a_i = a_i + \Delta_i$$

Where: $\Delta_i = \sum_{x,y}(\phi_i[x,y]I[x,y]) - \sum_{x,y}[\phi_i(\sum_{k=1}^{n} a_k\phi_k[x,y])]$

Or in other words, the difference between the product of the Gabor function with the original image and the Gabor function with the reconstructed image. The net is allowed to iterate until the adjustments, Δ_i, equal zero.

Put simply, the first layer computes the correlation of the image with the basis set. These coefficients are then used to reconstruct a new image $\sum_k[a_k\phi_k(x,y)]$. The third layer correlates the reconstructed image with the basis set. The result of the third layer correlation is compared to the correlation from the first layer and the adjustment to the coefficients is made. When the adjustment to each coefficient, Δ_i, is zero, the iteration process is terminated and the coefficients are read from the weights of the second layer. Overall, the function of the net is to calculate the amount of correction necessary to account for the non-zero inner products of each of the functions with all other functions as a result of the non-orthogonality of Gabor functions.

The initial implementation of the Gabor filters was as given in Daugman. The Gaussian envelope was supported on a 32 X 32 window with the $1/e$ point set at ± 9 pixels and tapered to a value of 0.05 at the windows edge where the value was truncated to zero. These 32 X 32 windows were moved across the image in steps of one-half the window size. So for the case of a 32 X 32 window the step size was 16 pixels.

Implementation of the network was done, but it failed to converge as described. Several modifications were made to the network and it still would not converge. The initial test of the Lenna image resulted in the network diverging rapidly while trying to find the optimal coefficients. Tests with other images did exactly the same thing.

It was later learned from a graduate student from University of Tennessee (13) that he had experienced the same problem while trying to implement Daugman's network. His fix had been to use only real images and the real (cosine) portion of the Gabor filters. In addition, his first estimate of the a_i coefficients had been the largest pixel value in the window of interest.

Appendix B. *Other RBF Segmentation Results*

Figure 29. Mission 85, Frame 27, RBF Segmentation, 5 x 5 Median Filter

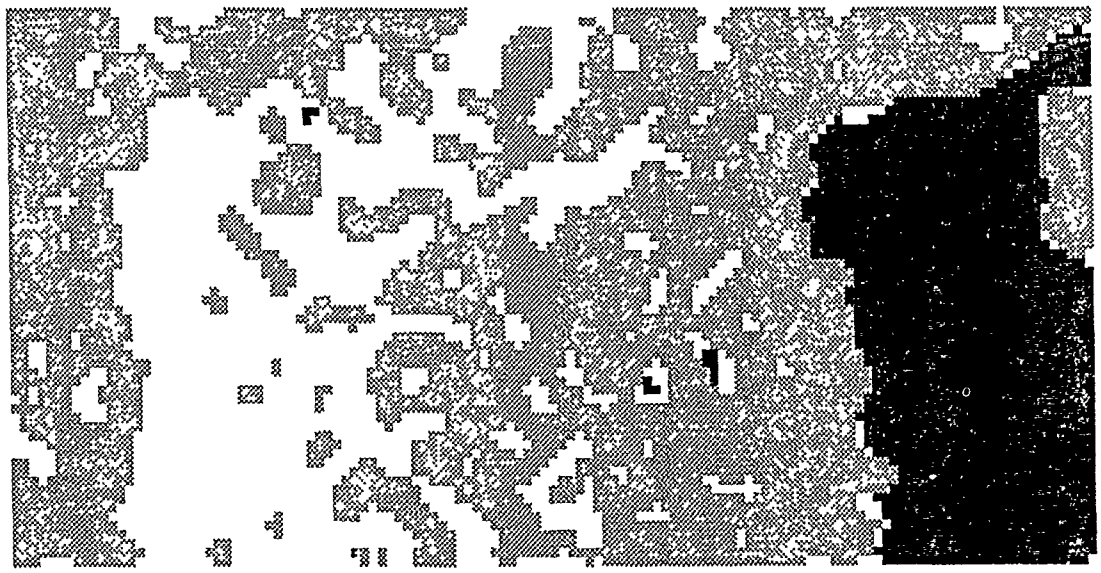

Figure 30. Mission 85, Frame 28, RBF Segmentation, 5 x 5 Median Filter

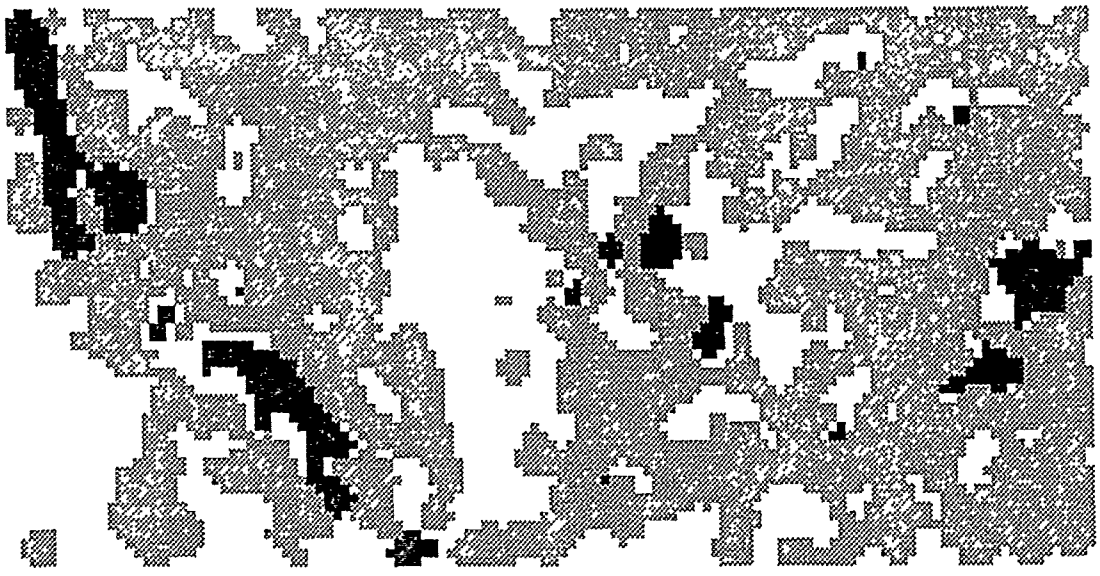

Figure 31. Mission 85. Frame 29, RBF Segmentation, 5 x 5 Median Filter

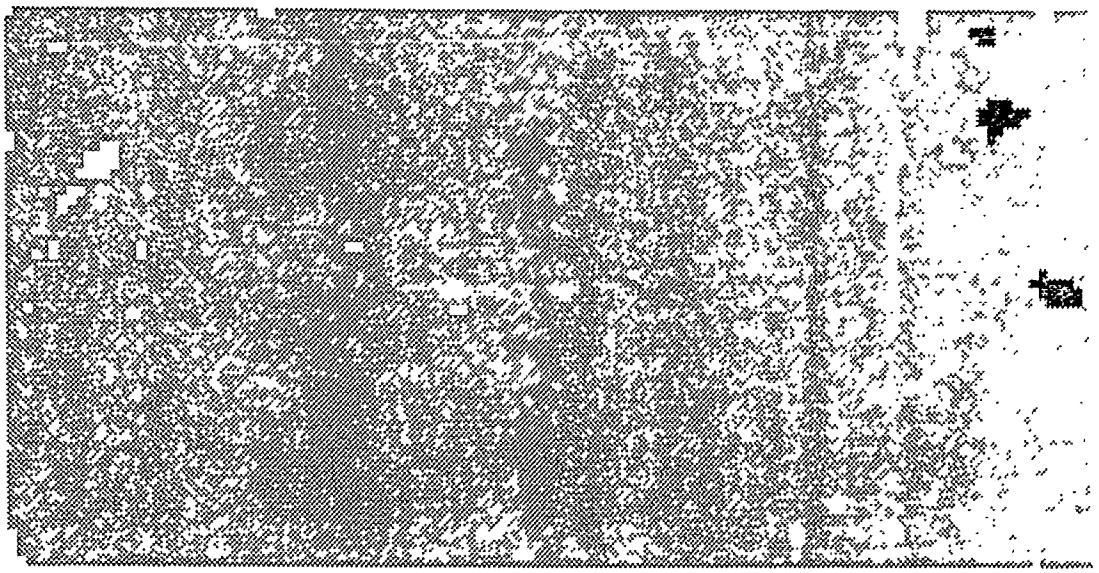

Figure 32. Mission 85, Frame 30, RBF Segmentation, 5 x 5 Median Filter

Figure 33. Mission 98, Frame 08, RBF Segmentation

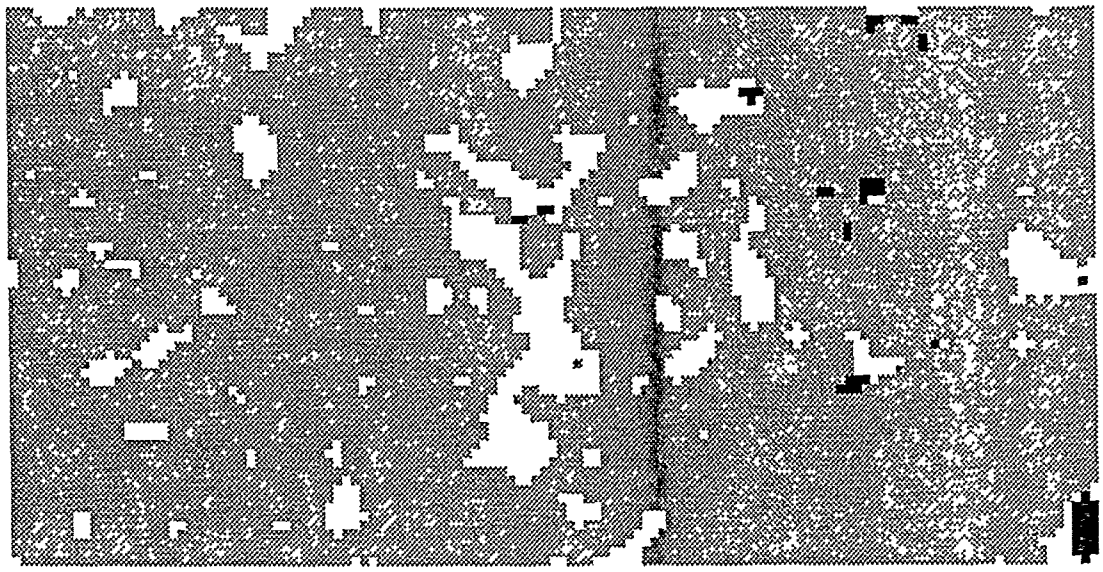

Figure 34. Mission 98, Frame 09, RBF Segmentation, 5 x 5 Median Filter

Appendix C. *Program Listings*

C.1 correlate.c

```
/*-------------------------------------------------------------*/
/* Program:  correlate.c                                       */
/*                                                             */
/* Author:  Michael Hazlett                                    */
/* Purpose:  Correlates an image with the given set of filters. */
/*-------------------------------------------------------------*/

#include <stdio.h>
#include <math.h>

#define TRUE 1
#define FALSE 0
#define SQR(a)  ((a)*(a))
#define SQR2(a) ((double)((a)*(a)))

void main()

{

float   **matrix(), *vector();
void     fourn(), free_vector(), free_matrix();

FILE   *fp, *gp, *hp;
int     numdim = 2, sign = 1, wsize, isize[2], size[2];
int     hwin, vwin, nwin, win;
int     xy, xsize, ysize, l, xr, xi;
int     ireal, nfilt;
int     winr, wini, winnum, filt;
int     row, col;
int     h, i, j, k, x, y, M, N;
char    gfile[60], cfile[60], ifile[60], cfile2[60], stype[2];
char    itype[2], dtype[2], dname[50], rmode[3], yn[2], cname[5];
float  **ai, **image, **gabor, **temp, *norms;
double  minval = 0.0000100;

 strcpy(rmode, "r");
```

```
printf("\nName of image file : ");
scanf("%s", ifile);

printf("\nIs the image [R]eal or [C]omplex : ");
scanf("%s", itype);

if(!(strcmp(itype, "R")) || !(strcmp(itype, "r"))) {
   printf("\nAre the numbers [F]loat or [I]nteger : ");
   scanf("%s", dtype);
}
else {
   printf("\nIs the file stored in [A]SCII or [I]nternal format?
: ");
   scanf("%s", stype);

   if(!(strcmp(stype, "I")) || !(strcmp(itype, "i"))) {
      strcpy(rmode,"rb");
   }
}

printf("\nSixe of image vertically (pixels) : ");
scanf("%d", &isize[0]);

printf("\nSize of image horizontally (pixels) : ");
scanf("%d", &isize[1]);

printf("\nWindow size of Gabor functions (pixels) : ");
scanf("%d", &wsize);

printf("\nNumber of filters : ");
scanf("%d", &nfilt);

printf("\nDirectory where filter files at : ");
scanf("%s", dname);

printf("\nName of coefficient file to create : ");
scanf("%s", cfile);

/*--------------------------------------*/
/* Initialize arrays and other variables */
/*--------------------------------------*/

M = wsize / 2;
xsize = isize[1] + M+1;
ysize = isize[0] + M+1;
```

```
 hwin  = isize[1] / M;
 vwin  = isize[0] / M;
 nwin  = hwin * vwin;

 image  = matrix(1, ysize, 1, 2*xsize);
 a1     = matrix(1, nfilt, 1, 2*nwin);
 gabor  = matrix(1, wsize, 1, 2*wsize);
 norms  = vector(1, nwin);

/*---------------------*/
/* Open the image file */
/*---------------------*/
 if ((fp = fopen(ifile, "r"))==NULL) {
    printf("Error opening file %s", ifile);
    exit(1);
 }

/*-----------------------------*/
/* Read in the image file data */
/*-----------------------------*/

/* Data stored is only real numbers */

 if(!(strcmp(itype, "r")) || !(strcmp(itype, "R"))) {

/* Real numbers, data values are integers */

    if(!(strcmp(dtype, "i")) || !(strcmp(dtype, "I"))) {

       for(i=1; i <= ysize; i++) {
          for(j=1; j <= xsize; j++) {
             xr = 2*j-1;
             xi = xr+1;
             if(i > M/2 && i <= ysize-M/2-1 &&
                j > M/2 && j <= xsize-M/2-1) {

                fscanf(fp, "%d\n", &ireal);
                image[i][xr] = (float)ireal;
                image[i][xi] = 0.0;
             }
             else {
                image[i][xr] = 0.0;
                image[i][xi] = 0.0;
             }
```

```
            }
         }
      }

   /* Real numbers, data values are float */

      else {

         for(i=1; i <= ysize; i++) {
            for(j=1; j <= xsize; j++) {
               xr = 2*j-1;
               xi = xr + 1;
               if(i > M/2 && i <= ysize-M/2-1 &&
                  j > M/2 && j <= xsize-M/2-1) {
                  fscanf(fp, "%f\n", &image[i][xr]);
                  image[i][xi] = 0.0;
               }
               else {
                  image[i][xr] = 0.0;
                  image[i][xi] = 0.0;
               }
            }
         }
      }
   }

   /* Data is stored as complex pairs */

    else {

   /* Complex pairs are stored in ASCII format */

       if(!(strcmp(stype, "A")) || !(strcmp(stype, "a"))) {

          for(i=1; i <= ysize; i++) {
             for(j=1; j <= xsize; j++) {
                xr = 2*j-1;
                xi = xr + 1;
                if(i > M/2 && i <= ysize-M/2-1 &&
                   j > M/2 && j <= xsize-M/2-1) {
                   fscanf(fp, "%f %f\n", &image[i][xr],
&image[i][xi]);
                }
                else {
                   image[i][xr] = 0.0;
```

```
                image[i][xi] = 0.0;
            }
        }
    }
}

/* Complex pairs are stored in INTERNAL format */

    else {

        temp = matrix(1, isize[0], 1, 2*isize[1]);

        for(i=1; i <= isize[0]; i++) {
             fread(temp[i], sizeof(float), 2*isize[1], fp);
        }

        for(i=1; i <= ysize; i++) {
           for(j=1; j <= xsize; j++) {
              xr = 2*j-1;
              xi = xr + 1;
              if(i > M/2 && i <= ysize-M/2-1 &&
                 j > M/2 && j <= xsize-M/2-1) {
                 k = 2*(j-M/2) - 1;
                 l = k + 1;
                 image[i][xr] = temp[i-M/2][k];
                 image[i][xi] = temp[i-M/2][l];
              }
              else {
                 image[i][xr] = 0.0;
                 image[i][xi] = 0.0;
              }
           }
        }

        free_matrix(temp, 1, isize[0], 1, 2*isize[1]);
    }

}

fclose(fp);

/*---------------------------*/
/* Zero out the coefficients */
/*---------------------------*/
 for(i=1; i <= nwin; i++) {
```

```
      for(j=1; j <= nfilt; j++) {
         a1[j][2*i-1] = 0.0;
         a1[j][2*i]   = 0.0;
      }
   }

/*---------------------------------------*/
/* Loop through all of the Gabor filters */
/*---------------------------------------*/
   for(i=1; i <= nfilt; i++) {

      sprintf(gfile, "%s%s%03d", dname, "gabor", i);

      if ((gp = fopen(gfile, "r"))==NULL) {
         printf("Error opening file %s", gfile);
         exit(1);
      }

      printf("\nGabor filter #%d", i);

/*------------------------*/
/* Read in the Gabor file */
/*------------------------*/
      for(j=1; j <= wsize; j++) {
         for(k=1; k <= wsize; k++) {
            xr = 2*k-1;
            xi = xr + 1;
            fscanf(gp, "%f %f\n", &gabor[j][xr], &gabor[j][xi]);
         }
      }
      fclose(gp);

/*-------------------------------------------------------------*/
/* Multiply the Gabor and Image point by point for each window */
/*-------------------------------------------------------------*/
      for(j=1; j <= vwin; j++) {

         for(k=1; k <= hwin; k++) {

            win = 2*((j-1)*hwin + k)-1;

            for(y=1; y <= wsize; y++) {

               row = (j-1)*(wsize/2)+y;
```

```
            for(x=1; x <= wsize; x++) {

               col = (2*(k-1)*(wsize/2))+2*x-1;

               a1[i][win] += image[row][col] * gabor[y][2*x-1] -
                             image[row][col+1] * -gabor[y][2*x];
               a1[i][win+1] += image[row][col+1] * gabor[y][2*x-1]
                             + image[row][col]   * -gabor[y][2*x];

            }
         }
      }
   }
}

/*-----------------------------------------------*/
/* Store the original coefficients in the file. */
/* Store by window # then filter #. Normalize   */
/* coefficients to area under Gabor.            */
/*-----------------------------------------------*/
 if ((hp = fopen(cfile, "w"))==NULL) {
    printf("Error opening file %s", cfile);
    exit(1);
 }

 fprintf(hp, "%d %d %d %d %d %d\n", nfilt, wsize, hwin, vwin,
         isize[0],isize[1]);

 for(i=1; i <= nwin; i++) {
    for(j=1; j <= nfilt; j++) {
       if(fabs((double)a1[j][2*i-1]) < minval) a1[j][2*i-1] = 0.0;
       if(fabs((double)a1[j][2*i]) < minval) a1[j][2*i] = 0.0;
       fprintf(hp, "%d %d %e %e\n",i, j, a1[j][2*i-1], a1[j][2*i]);
    }
    fprintf(hp, "\n");
 }

 fclose(hp);

 printf("\nFinished computing a1 coefficients\n");

}
```

C.2 rebuild.c

```
/*-------------------------------------------------------*/
/*  Program rebuild.c                                    */
/*                                                       */
/*  Author:  Michael Hazlett                             */
/*  Purpose: Creates an image from a file containing     */
/*           coefficient values                          */
/*-------------------------------------------------------*/

#include <stdio.h>
#include <math.h>

void main()

{

float  **matrix();
int    *ivector();

FILE   *fp, *gp;
int     wsize, isize[2], xsize, ysize, win, f1, f2;
int     hwin, vwin, nwin, xpos, ypos;
int     nfilt, xstart, ystart, w, *filt, f;
int     i, j, k, x, y;
char    gfile[60], cfile[32], ifile[32], gname[50], yn[2];
char    dname[30];
float  **gabor, **image, **ai;

 printf("Name of coefficient file : ");
 scanf("%s", cfile);

 printf("\nName of image file to create : ");
 scanf("%s", ifile);

 printf("\nDirectory where Gabor files at : ");
 scanf("%s", dname);

 /*--------------------------*/
 /* Open the coefficient file */
 /*--------------------------*/
  if((fp=fopen(cfile, "r"))==NULL) {
     printf("\nError opening %s\n", cfile);
```

```
    exit(1);
 }

/*-----------------------------*/
/* Read in reconstruction info */
/*-----------------------------*/
 fscanf(fp, "%d %d %d %d %d %d", &nfilt, &wsize, &hwin, &vwin,
&isize[0], &isize[1]);

/*-----------------------------------*/
/* Initialize arrays and other variables */
/*-----------------------------------*/
 nwin  = hwin * vwin;
 ysize = isize[0]+wsize/2+1;
 xsize = 2*(isize[1]+wsize/2+1);

 filt = ivector(1, nfilt);

 image = matrix(1, ysize, 1, xsize);
 gabor = matrix(1, wsize, 1, 2*wsize);

 ai   = matrix(1, nfilt, 1, 2*nwin);

 sprintf(gname, "%s%s", dname, "gabor");

/*----------------------*/
/* Zero the image array */
/*----------------------*/
 for(i=1; i<= ysize; i++) {
    for(j=1; j<=xsize; j++) {
       image[i][j] = 0.0;
    }
 }

/*-------------------------------------------------*/
/* Initialize filter selection array to all filters */
/*-------------------------------------------------*/
 for(i=1; i<=nfilt; i++) {
    filt[i] = i;
 }

/*--------------------------*/
/* Read in the coefficients */
/*--------------------------*/
 for(i=1; i <= nwin; i++) {
```

```
   for(j=1; j <= nfilt; j++) {
      fscanf(fp, "%d %d %f %f\n", &w, &f, &ai[j][2*i-1],
             &ai[j][2*i]);
   }
   fscanf(fp, "\n");
}

fclose(fp);

printf("\nRebuild the image from selected filter(s)? (y/n) : ");
scanf("%s", yn);

if(!(strcmp(yn, "y")) || !(strcmp(yn, "Y"))) {

   i = 1;

   while(f && i <= nfilt) {
     printf("\nFilter # (0 to stop) : ");
     scanf("%d", &f);
     filt[i] = f;
     i++;
   }
}

i = 1;

/*-----------------------------------------*/
/* Loop through all of the Selected filters */
/*-----------------------------------------*/
 while(filt[i] && i <= nfilt) {

    f = filt[i];

    sprintf(gfile, "%s%03d", gname, f);

    if ((gp = fopen(gfile, "r"))==NULL) {
       printf("Error opening file %s", gfile);
       exit(1);
    }
/*------------------------*/
/* Read in the Gabor file */
/*------------------------*/
    for(j=1; j <= wsize; j++) {
       for(k=1; k <= wsize; k++) {
          fscanf(gp, "%f %f\n", &gabor[j][2*k-1], &gabor[j][2*k]);
```

```
        }
      }
      fclose(gp);

/*-------------------------------------------------------*/
/* Multiply each of the coefficient by the Gabor at each  */
/* location and sum the result to reconstruct the image.  */
/*-------------------------------------------------------*/
      for(j=1; j <= vwin; j++) {

         for(k=1; k <= hwin; k++) {

            win = 2*((j-1)*hwin + k)-1;

            for(y=1; y <= wsize; y++) {

               ypos = (j-1)*(wsize/2)+y;

               for(x=1; x <= wsize; x++) {

                  xpos = (2*(k-1)*(wsize/2))+2*x-1;

                  image[ypos][xpos] += ai[f][win] * gabor[y][2*x-1]
                                     -  ai[f][win+1] * gabor[y][2*x];
                 image[ypos][xpos+1] += ai[f][win+1] * gabor[y][2*x-1]
                                     + ai[f][win]   * gabor[y][2*x];

               }
            }
         }
      }

      i++;
   }

/*-------------------------------------------------*/
/* Write the reconstructed image array to a file */
/*-------------------------------------------------*/
   if((fp=fopen(ifile, "w"))==NULL) {
      printf("Error opening %s\n", ifile);
      exit(1);
   }
   ystart = (wsize/4) + 1;
   xstart = (wsize/4) + 1;
   for(i=ystart; i <= ystart+isize[0]-1; i++) {
```

```
    for(j=xstart; j <= xstart+isize[1]-1; j++) {
       fprintf(fp, "%f %f\n", image[i][2*j-1], image[i][2*j]);
    }
 }

  fclose(fp);

}
```

Bibliography

1. Ayer, Kevin W. *Gabor Transforms for Forward Looking InfraRed Image Segmentation*. MS thesis, AFIT/GE/ENG/89D-1, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1989.

2. Bovik, Alan Conrad and others. "Multichannel Texture Analysis Using Localized Spatial Filters," *IEEE PAMI*, *12*(1) (January 1990).

3. Daughman, John G. "Complete Discrete 2-D Gabor Transforms by Neural Networks for Image Analysis and Compression," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, *36*(7):1169–1179 (July 1988).

4. Daugman, John G. "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *Journal of the Optical Society of America*, (7):1160–1169 (July 1985).

5. Jones, P. Judson and Larry A. Palmer. "An Evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex," *Journal of Neurophysiology*, *58*(6):1233–1258 (December 1987).

6. L'Homme, Albert P. *Gabor Filters and Neural Networks for Segmentation of Synthetic Aperture Radar Imagery*. MS thesis, AFIT/GE/ENG/90D-31, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, 1990.

7. Lu, Shin-yee and others. "Texture Segmentation by Clustering of Gabor Feature Vectors," *International Joint Conference on Neural Networks, Seattle*, *I*:683–687 (July 1991).

8. Novak, L. M. and others. "Optimal Processing of Polarimetric Synthetic Aperture Radar Imagery," *The Lincoln Laboratory Journal*, *3*(2) (1990).

9. Pierce, J. R. and J. E. Carlin. "Reading Rates and the Information Rate of a Human Channel," *Bell Systems Technical Journal*, *36*:497–516 (March 1957).

10. Porat, Moshe and Yehoshua Y. Zeevi. "The Generalized Gabor Scheme of Image Representation in Biological and Machine Vision," *IEEE PAMI*, *10*:452–468 (July 1988).

11. Priddy, Kevin L., et al. "Image Segmentation, Feature Extraction, and Classification Using Neural Networks." *Proceedings of the Sixth Annual Aerospace Applications of Artificial Intelligence Conference*. ACM Press, 1990.

12. Ruck, Dennis W. *Characterization of Multilayer Perceptrons and their Application to Multisensor Automatic Target Detection.* PhD dissertation, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, 1990.

13. Sarif-Saref, Hamed. Personal Correspondance. University of Tennessee, Knoxville TN, 1991.

14. The Khoros Group. *Khoros Manual Volume I User's Manual.* Albuquerque NM: University of New Mexico, May 1991.

15. Veronin, Christopher P., et al. "An optical image segmentor using wavelet filtering techniques as the front end of a neural network classifier." *Proceedings of the SPIE 1991 International Symposium and Exihibition on Optical Engineering and Photonics in Aerospace Sensing.* Bellingham, WA: SPIE Press, 1991.

16. Wong, Yiu-fai. "How Gaussian Radial Basis Functions Work," *IJCNN, Seattle, II*:133–137 (July 1991).

17. Zahirniak, Daniel R. *Characterization of Radar Signals Using Neural Networks.* MS thesis, AFIT/GE/ENG/90D, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, 1990.

18. Zahirniak, Daniel R and others. "Pattern Recognition using Radial Basis Function Networks." Prepared for AIAAA Conference.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1991	Master's Thesis

4. TITLE AND SUBTITLE
Gabor Segmentation of High Resolution Synthetic Aperture Radar Imagery

5. FUNDING NUMBERS

6. AUTHOR(S)
Michael A. Hazlett, Captain, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GE/ENG/91D-23

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This thesis investigates the use of Gabor filters and a radial basis function (RBF) network for segmentation of high resolution (1 foot by 1 foot) synthetic aperture radar (SAR) imagery. Processing involved correlation between the SAR imagery and Gabor functions. Two methods for selecting the optimal Gabor filters are presented. This research used complex Gabor functions and operated on single polarization HH complex data. Following the selection of the proper Gabor filters, correlation coefficients for each image were calculated and used as features for the RBF network. Provided are results demonstrating how Gabor processing and a RBF network provide image segmentation.

14. SUBJECT TERMS
Gabor Functions, Image Analysis, Segmentation

15. NUMBER OF PAGES
75

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to *stay within the lines* to meet *optical scanning requirements.*

Block 1. Agency Use Only *(Leave blank)*

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	- Contract	PR	- Project
G	- Grant	TA	- Task
PE	- Program Element	WU	- Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

- DOD - See DoDD 5230.24, "Distribution Statements on Technical Documents."
- DOE - See authorities.
- NASA - See Handbook NHB 2200.2.
- NTIS - Leave blank.

Block 12b. Distribution Code.

- DOD - Leave blank.
- DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
- NASA - Leave blank.
- NTIS - Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

*U.S. GPO 1989-273-271